Modelling vortex motion with a stream function subdivision finite element

Fisher, Alex John

Award date:
2007

Awarding institution:
University of Bath

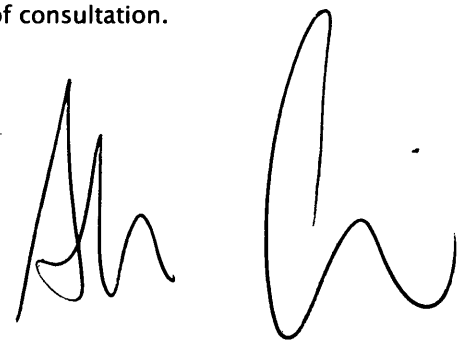
Modelling vortex motion with a stream function subdivision finite element

Alex John Fisher

A thesis submitted for the degree of
Doctor of Philosophy

University of Bath
Department of Architecture and Civil Engineering

October 2007



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

This thesis develops an approach for discretising and solving the Navier-Stokes equations characterising viscous fluid flow. This subject of describing partial differential equations is approached through the analogous field of computer-aided geometric design. Recent developments in efficient techniques for surface definition used in computational design and computer animation are presented.

Focus is placed on the generation of the complex surface geometries describing the vortex formations occurring during fluid flow past a bluff body at high Reynolds numbers.

The Navier-Stokes equations are written in a two-dimensional stream function formulation. This enables the analogue with plate bending to be observed. A curvilinear subdivision finite element is employed, satisfying the required C^1 continuity on an arbitrary topology mesh.

The conformity of the implementation is initially validated using analytic structural and Stokes flow solutions. The dissertation then focuses on modelling fluid flow past a circular cylinder for a large range of Reynolds numbers. Re = 40, 100, 250, 1000 and 4500 are modelled, successfully reproducing characteristic vortex behaviour with an accurate correlation to existing computational and experimental results. The high Reynolds number simulations are seen to produce an intricate, almost fractal-like hierarchy of vortices in the near wake of the cylinder. Applications and limitations of this two-dimensional model are discussed.

Acknowledgements

I would firstly like to thank Chris Williams for his ongoing support and enthusiasm for my research. I have really thoroughly enjoyed working with him.

There are a number of other people whose involvement I have greatly appreciated. I especially wish to thank Lisa Matthews for her significant contribution. It has been great fun working together. Also very importantly, thanks go to Jon, Fred, Roly and Jan. Without them writing this thesis would have been a lot harder, and a lot less entertaining.

I would also like to thank my whole family, in particular my parents, and then finally Laura. Laura's support and encouragement has been immense and I would not have liked to have attempted this without her.

Contents

List of figures and tables

Nomenclature

The Einstein summation convention is adhered to throughout this dissertation. Here the summation is implied over all values of a repeated index occurring exactly once in each of the superscript and subscript positions. Latin lower case indices generally take the values 1-3 and Greek lower case indices 1 and 2.

In part three Greek upper case letters for indices denote shape functions and Latin upper case letters global degrees of freedom.

$\mathbf{a}_i, \mathbf{a}^i$	covariant and contravariant base vectors of the surface
$a_{\alpha\beta}, a^{\alpha\beta}$	metric tensors of the surface; coefficients of the first fundamental form
a	determinant of $a_{\alpha\beta}$
$\mathbf{A}_i, \mathbf{A}^i$	covariant and contravariant base vectors of the surface in the reference configuration
$A_{\alpha\beta}, A^{\alpha\beta}$	surface metrics in the reference configuration
$b_{\alpha\beta}, b^{\alpha\beta}$	coefficients of the second fundamental form
$C^{\Pi\Theta\Sigma}$	convection matrix
d	cylinder diameter
$D^{\Pi\Theta}$	damping matrix
D	plate stiffness
E	Young's modulus of elasticity
f_I	nodal residual forces
F_I	convection and stiffness forces

$\mathbf{g}_i, \mathbf{g}^i$	covariant and contravariant base vectors
g_{ij}, g^{ij}	metric tensors
g	determinant of g_{ij}
g	number of holes in a closed surface (genus)
$\mathbf{G}_i, \mathbf{G}^i$	covariant and contravariant base vectors in the reference configuration
G_{ij}, G^{ij}	metric tensors in the reference configuration
G	determinant of G_{ij}
H	mean curvature
$\mathbf{i}_k \equiv \mathbf{i}^k$	Cartesian unit base vectors
k_0, k_1, k_2	valance of element's nodes 0, 1 and 2 respectively
$K^{\Pi\Theta}$	stiffness matrix
K	Guassian curvature
$m^{\alpha\beta}$	stress couple
$m_{(\alpha\beta)}$	physical stress couple
$\mathbf{n}_\alpha$	stress resultants on the θ^α -surface
$n^{\alpha\beta}$	stress resultant
$n_{(\alpha\beta)}$	physical stress resultant
n	maximum number of subsidiary iterations
N^Π	subdivision shape functions
q^α	shearing force
$q_{(\alpha)}$	physical shearing force
p	transverse loading per unit area
$\mathbf{r}$	shell position vector
r	radius

$\mathbf{R}$	shell position vector in the reference configuration
Re	Reynolds number $\frac{\rho V d}{\mu}$
S_{Π}^{P}	local subdivision matrix
S	surface area
St	Strouhal number
$\mathbf{t}_{\alpha}$	stress vector per unit area, associated with $\mathbf{n}_{\alpha}$
t	time; thickness
t^{*}	nondimensional time
δt	subsidiary time step
ΔT	time step in temporal loop
T	period of oscillation
$\mathbf{T}_{\alpha}$	force acting on the θ^{α} -surface
V	characteristic velocity
W^{I}	weighting functions
$\mathbf{x}_{P}, \mathbf{x}_{\Pi}$	global and local element control points
$\mathbf{x}$	surface position vector
x^{0}	time (chapter 3a)
$\mathbf{X}$	surface position vector in the reference configuration
x^{i}, X^{i}	Cartesian coordinates in the current and reference configurations
β_{one}, β_{two}	ghost mirroring constants
$\gamma_{\alpha\beta}, \gamma^{\alpha\beta}$	strain tensors
Γ_{ij}^{r}	Christoffel symbols of the second kind
$\delta_{ij}, \delta^{ij}, \delta_{j}^{i}$	Kronecker deltas

$\varepsilon_{rst}, \varepsilon^{rst}, \varepsilon_{\alpha\beta}, \varepsilon^{\alpha\beta}$	permutation symbols
ζ	viscous damping relaxation factor
θ^i	curvilinear coordinates
θ_s	separation angle
κ	curvature
μ	viscosity
ν	Poisson's ratio
ξ	carry over dynamic relaxation factor
ρ	density
$\tau^{\alpha\beta}$	stress tensor
ϕ	equipotentials
χ	Euler characteristic of the surface
ψ	stream function; transverse plate deflection
$\omega_{\alpha\beta}$	vorticity tensor
∇_α	covariant derivative

Part one

Research context

Chapter one

Introduction

1.0 Background

This thesis is concerned with generating complex curved surface geometries. The problem of interest is the class of surfaces describing vortex motion through viscous fluids. However, in general the challenge of describing curved surfaces is widespread as the term *surface* can be taken to mean various things in different contexts. Essentially, here a surface is defined as any continuum, geometrical, physical or mathematical, which can be defined using exactly two coordinates. It is therefore two-dimensional, but with the possibility of being curved in three-dimensional space. This definition therefore includes the obvious geometrical surfaces such as the plane, sphere and polyhedra. However one can also define a second class of physical surfaces. These are entities whose thickness can be deemed negligible with regard to their other dimensions and so they can be idealised as a single two-dimensional surface. Numerous examples can be

imagined, taken from both the natural and engineered world. For instance, the shells of turtles, gastropods and eggs, soap films, spiders' webs and also concrete shells, masonry vaults, pressure vessels, balloons, fabrics, tensile membranes *et cetera*.

The third class of surfaces which is of particular interest in this thesis are mathematical functions of two variables, or differential equations expressing a two dimensional field in some variable. In part three focus is placed on the complex surfaces describing the stream function for incompressible fluid flow in two dimensions.

The topic of defining surfaces is therefore vast and spans many seemingly disparate disciplines. However generalisations can be made and useful parallels drawn. The central challenge in any surface description lies in deciding what function or rules govern the surface geometry for the given application. There are two potential options open to defining a continuum. In special cases an exact analytic solution may be known, describing the surface in its entirety.

1.0.1 Analytical surface description

Equation [1.1] is a simple example of a surface expressed analytically, visualised in Figure 1.1.

$$\begin{aligned} x &= e^{\theta \tan \lambda}\left(x_0 + r_0 \cos\phi\right)\cos\theta, \\ y &= e^{\theta \tan \lambda}\left(x_0 + r_0 \cos\phi\right)\sin\theta, \\ z &= e^{\theta \tan \lambda}\left(z_0 + r_0 \sin\phi\right). \end{aligned} \qquad [1.1]$$

$$x_0, z_0, r_0, \lambda = \text{constant}.$$

The surface in Figure 1.1 is formed as a circular cross section is swept along a logarithmic spiral (parameterised by ϕ and θ respectively). The advantage of expressing geometry analytically is that the solution is fully determined and *smooth* since derivatives are determined. There are therefore no discontinuities in slope or curvatures.

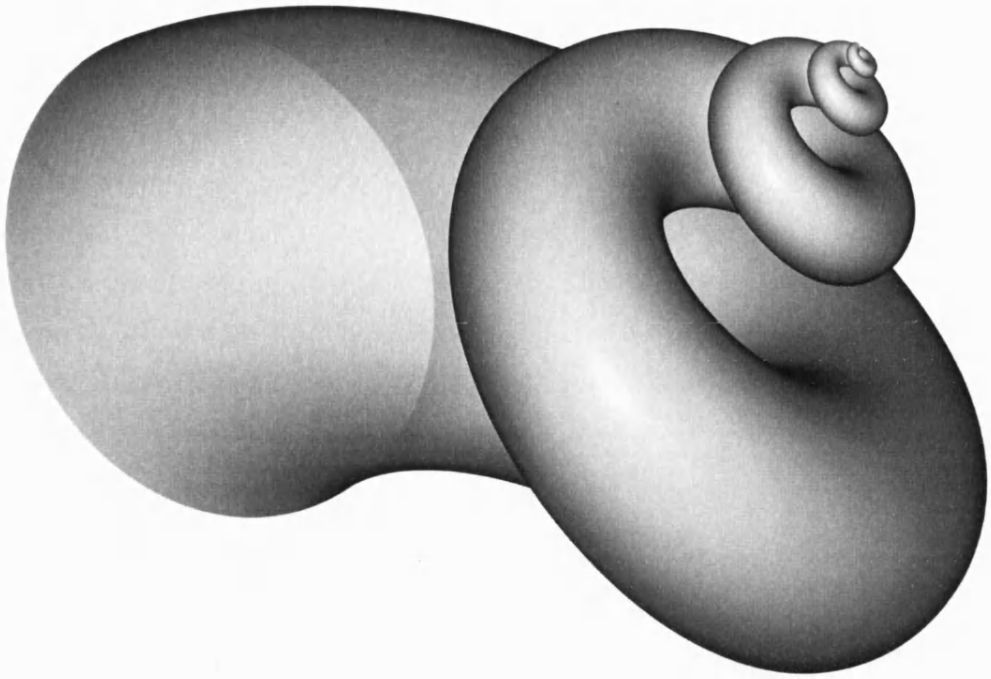


Figure 1.1 Analytical surface as defined in equation [1.1]

As Williams[1] says the complexity of the geometry that can be described in this way is limited only by one's mathematical knowledge. For increasingly complex problems this analytical solution may not therefore be known and alternative approaches to describing geometry must be taken through processes of approximation and discretisation.

This problem of expressing complex geometry was encountered in the automotive and aerospace industries in the 1960's when describing the curved surfaces integral to the design of cars and aeroplanes. In order to produce these forms a sculptural approach was taken. Traditionally, three-dimensional work was performed using clay models and, in two-dimensions, free-form curves were defined using flexible strips of wood called *splines*. Bent using weights named ducks[2], the elasticity of the spline creates a smooth curve of minimum energy. Alternatively when exact curves were required, designers used standard curve templates called French curves consisting of sections of conics and spirals pieced together.

1.0.2 Computer-aided geometric design

Following a drive for automation in design, computer modelling and manufacturing, the discipline of computer-aided geometric design developed. Here an approximate solution to a surface is described using some form of discretisation. An approximation or interpolation is performed filling in the gaps in the continuum between a number of known values or control points on the surface. The geometry can therefore be considered as broken up into a number of discrete patches forming a surface mesh. Continuous complex geometries can then be easily generated by manipulating the finite number of control points. An example application of a mixed sculptural- and analytical-approach to surface definition is given in Figure 1.2 taken from the paper *Biomimesis and the geometric definition of shell structures in architecture*[3] included as an appendix to this thesis.

The smoothness of a discretisation is controlled by the continuity between the elements. A curved surface can be discretised in to simple planar facets with basic geometrical connectivity at the interface between elements. Alternately smoothness can be improved by increasing the constraints enforced on the surface derivatives, imposing continuity of slope, curvature *et cetera*. The level of continuity required will depend on the specific application. As Sederberg[4] describes, continuous curvature is required for the design of rail tracks to prevent the train experiencing sudden jarring. A similar requirement is necessary for smooth rendering of reflection lines on surfaces such as car bodies. In general however the surface description required will be less restrictive than fully continuous curvature, with specific applications demanding conformity of slope or only simple geometric connectivity between elements.

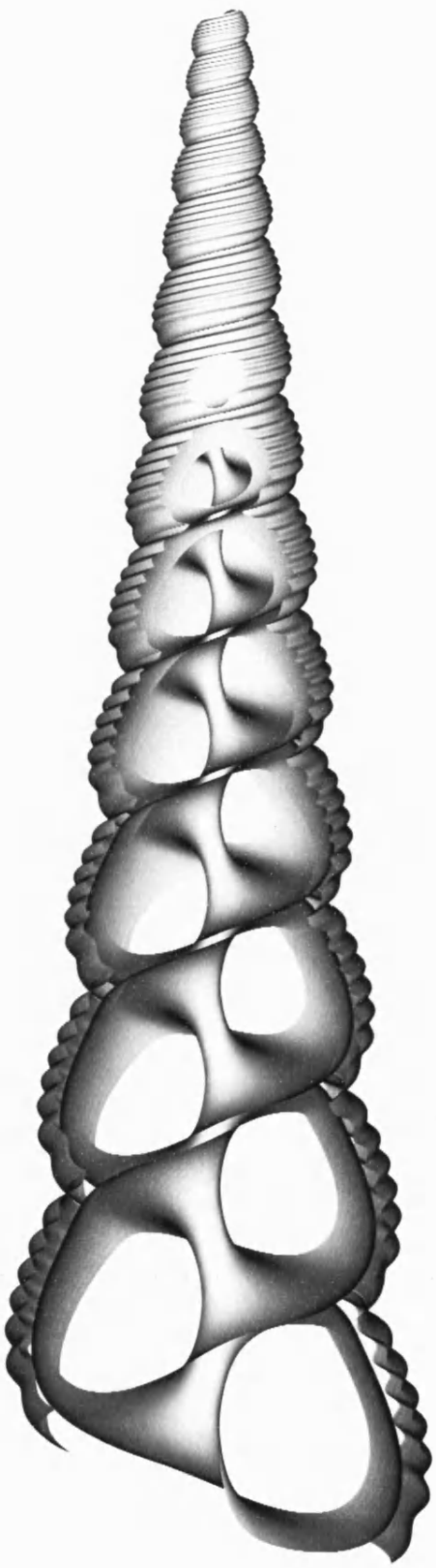


Figure 1.2 Parametric sculptural surface, generated from the algorithm presented in Appendix A

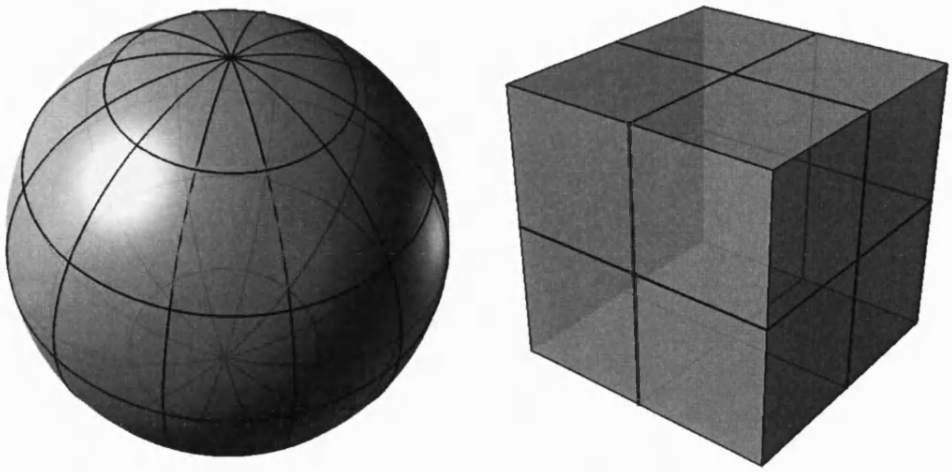


Figure 1.3 Extraordinary points occurring in square curvilinear grids on surfaces homeomorphic to a sphere

The challenge that arises is to define a suitable control mesh with the necessary continuity that can accurately, but efficiently, characterise the surface. Computer-aided design techniques have historically favoured rectangular meshes. This is largely because surface definition has traditionally been thought of as an extrapolation of curve methodologies forming a surface as a tensor product of two curves. However Farin[5] also notes the natural geometry of car body panels, roof and doors are curvilinear rectangular. This led to many of the procedures being intrinsically embedded in a bidirectional parameterisation. It will be shown that this can be problematic when dealing with irregular topology meshes.

Irregularity in a mesh can be enforced due to the topological nature of the surface as shown in Figure 1.3. This is a consequence of the fact that any continuous tangent vector field on a topologically spherical surface must have a point where the vector is zero†[6]. This is not true for all surfaces. A regular mesh can be mapped on to a surface of toroidal topology for example. However for certain closed surfaces the consequence is there may be a requirement for extraordinary points in the control mesh.

† This concept, sometimes known as the *hairy ball theorem*, can be visualised by the tufts of fur found on a dog's stomach. Here the hairs can no longer lie flat, as on the dog's back, and are forced to stand normal to the dog's body.

Distributing control points for efficiency is the principle cause for irregularity in a mesh. Vertices and elements may be defined responding to the level of geometrical detail required in different regions of the discretisation as illustrated in Figure 1.4. Flat and featureless areas of a surface can naturally be defined with fewer control points. These issues will become important and shall be revisited in later chapters.

1.0.3 Numerical generation of surfaces

This methodology of discretising a geometrical surface can equally be applied to the problem of solving differential equations. Exact analytical solutions to certain classes of partial differential equations can be written. However more often than not, due to the complexity of either the equations, or the boundary conditions, or both, an analytical solution is either unknown or does not exist. In these circumstances an approximate solution must be sought numerically. In an analogous approach to the methods of computer-aided geometrical design the solution is subdivided into elements (or patches) and the infinite degrees of freedom are limited to a finite number. This creates a set of simultaneous algebraic equations which can be solved computationally. This process is called the finite element method and will be applied in detail in chapter six of this thesis.

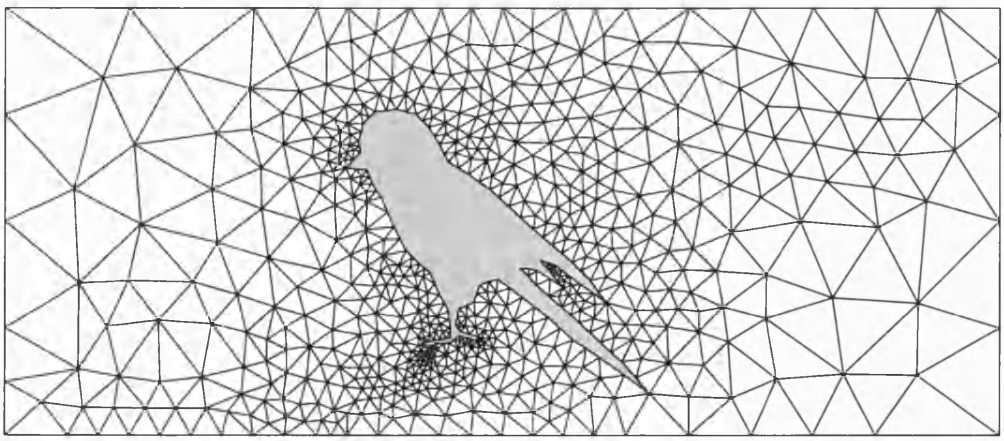


Figure 1.4 Irregular arbitrary topology mesh conforming to complex boundary geometry

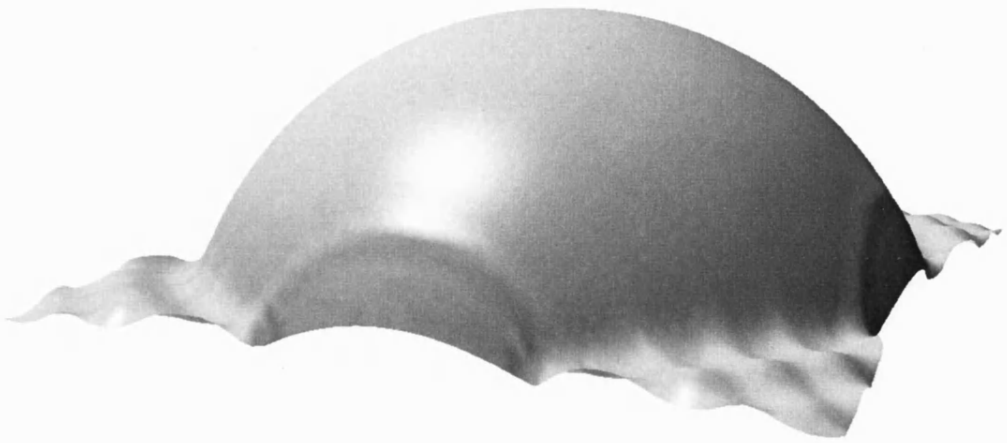


Figure 1.5 Surface generated computationally employing a dynamic relaxation solution procedure for a simple mass-spring discretisation (see section 6.5)

Complex surface geometries can then be generated, emerging naturally from the solution of the underlying equations. An example is given in Figure 1.5. It is this topic of discretising and solving differential equations using computational geometry techniques that forms the subject of this thesis.

The equations of interest are the incompressible Navier-Stokes equations describing fluid motion. They are given for three dimensions in Equation [1.2]. An explanation of their derivation will be given in later chapters. However from qualitative inspection they can be seen to be highly nonlinear and thus their solution for all but the simplest cases is difficult.

$$\begin{aligned}
\rho\left(\frac{\partial v_x}{\partial t}+v_x\frac{\partial v_x}{\partial x}+v_y\frac{\partial v_x}{\partial y}+v_z\frac{\partial v_x}{\partial y}\right)&=-\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 v_x}{\partial x^2}+\frac{\partial^2 v_x}{\partial y^2}+\frac{\partial^2 v_x}{\partial z^2}\right),\\
\rho\left(\frac{\partial v_y}{\partial t}+v_x\frac{\partial v_y}{\partial x}+v_y\frac{\partial v_y}{\partial y}+v_z\frac{\partial v_y}{\partial y}\right)&=-\frac{\partial p}{\partial y}+\mu\left(\frac{\partial^2 v_y}{\partial x^2}+\frac{\partial^2 v_y}{\partial y^2}+\frac{\partial^2 v_y}{\partial z^2}\right),\\
\rho\left(\frac{\partial v_z}{\partial t}+v_x\frac{\partial v_z}{\partial x}+v_y\frac{\partial v_z}{\partial y}+v_z\frac{\partial v_z}{\partial y}\right)&=-\frac{\partial p}{\partial y}+\mu\left(\frac{\partial^2 v_z}{\partial x^2}+\frac{\partial^2 v_z}{\partial y^2}+\frac{\partial^2 v_z}{\partial z^2}\right),\\
\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}&=0,
\end{aligned} \qquad [1.2]$$

v_x, v_y, v_z are the three velocity components associated with the x, y and z directions, p is pressure, ρ is the density and μ is the viscosity of the fluid.

The nonlinearity of the equations causes chaotic and intricate flow patterns to emerge. This characteristic of fluid flow is not a recent discovery. Figure 1.6 shows a sketch by Leonardo Da Vinci, produced at the start of the 16th century. The figure clearly shows his interpretation of the complex eddies and currents formed during fluid motion. A self similarity and fractal-like nature of the vortices can be observed in his illustration. Da Vinci likened these patterns to ringlets of hair[7].

The Navier-Stokes equations, were originally written by Navier in 1827 and then more succinctly by Stokes in 1845[8]. However the phenomenal complexity of these flows has meant that a full understanding providing accurate solutions to fluid behaviour remains a constant challenge. Analytic solutions exist for only very simple test cases. Thus numerical techniques must be employed to provide insight into the behaviour of problems of practical interest.

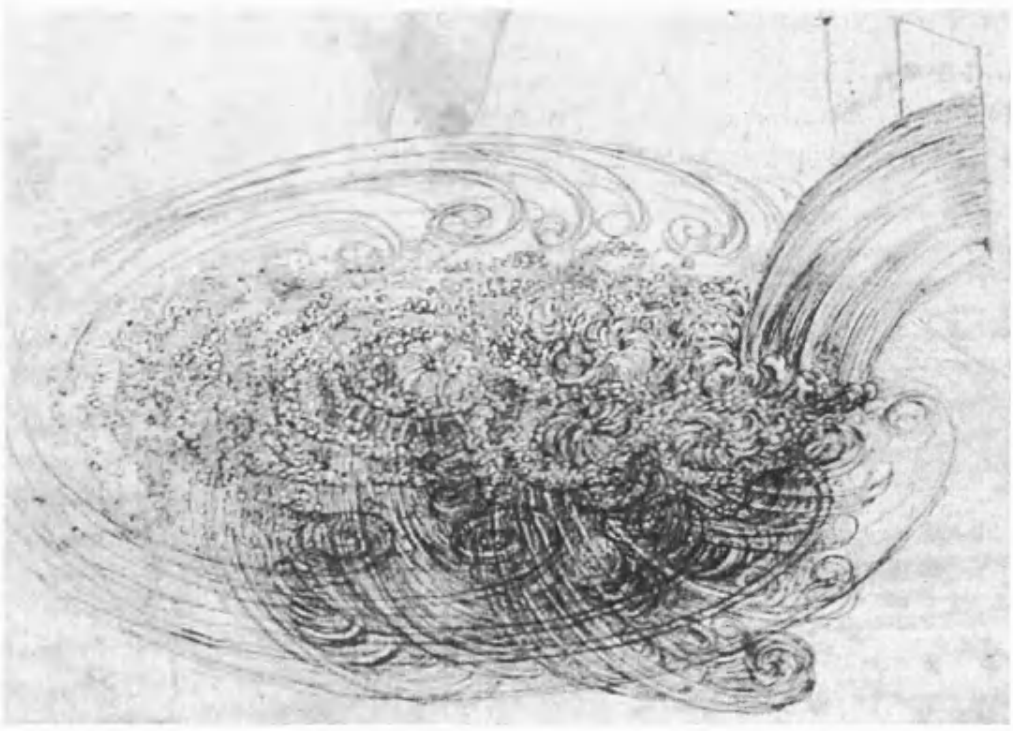


Figure 1.6 Illustration of vortex motion by Leonardo Da Vinci entitled *Sketch of a whirlpool*[9]

1.0.4 The stream function surface

In this thesis the Navier-Stokes equations are reformulated in a two-dimensional surface form using the stream function ψ. The stream function, whose gradient gives the velocity of the fluid, is defined in full in chapter four along with the full derivation of the following relationships. It is shown that the two-dimensional form of the equations in [1.2] can be rewritten as

$$\nabla^4\psi = \frac{\rho}{\mu}\left(\frac{\partial}{\partial t}\left(\nabla^2\psi\right)+\frac{\partial\psi}{\partial y}\frac{\partial}{\partial x}\left(\nabla^2\psi\right)-\frac{\partial\psi}{\partial x}\frac{\partial}{\partial y}\left(\nabla^2\psi\right)\right). \qquad [1.3]$$

Conceptually abstract mathematical surfaces can be visualised through analogy with more tangible physical surfaces. A useful structural analogy is observed when comparing the stream function [1.3] with the equation for plate bending used in structural engineering,

$$\nabla^4\psi = \frac{p}{D}, \qquad [1.4]$$

where ψ is the transverse deflection, p is the load and D is the stiffness constant.

For example, the case of Stokes flow in a channel (very low Reynolds number flow) is mathematically identical to the problem of a clamped plated under a transverse displacement at its boundary, where the right-hand side can be equated to zero in both equations [1.3] and [1.4].

Inspection of these equations shows that we are dealing with a fourth-order partial differential equation. It will be shown in chapter six that once discretised the equations contain second-order derivatives, the continuous evaluation of which requires conformity of the first derivative at the interface between surface elements. In computer-aided design this is referred to as C^1 continuity.

The discretisation and solution of these equations forms the subject of this research.

1.1 Aim, scope and objectives

1.1.1 Aim

The aim of this thesis is to formulate an approach for solving highly non-linear high order differential equations in two dimensions thus generating interesting and complex surface geometries.

1.1.2 Scope

Emphasis is placed with the stream function formulation of the Navier-Stokes equations of viscous fluid motion. Intrinsically complex surface geometries shall emerge naturally as a result of the inherent instabilities in the equations. The test case of flow past a circular cylinder is chosen as it constitutes a well observed multi-scale problem. An intricate hierarchical formation of vortices is seen to evolve in the boundary layer.

To efficiently model this behaviour a highly irregular underlying mesh is employed, changing in resolution around the regions of primary interest. Recent developments in surface representation initially developed for computer animation and computer gaming industries are therefore introduced and used as the discretisation of the stream function solution.

The efficient subdivision approach handles irregular topology control meshes whilst providing the high order of continuity required between elements. This existing approach to modelling surfaces is extended to solve the highly non-linear situation of viscous fluid flow past a circular cylinder. Reproduction of the small perturbations in the surface describing the vortex motion is thus enabled.

This thesis is concerned with generating complex surface geometries and thus focuses on two-dimensional problems, with the extension to three-dimensions comprising possible further work. The challenge is to provide a solution approach for forms of the biharmonic equation. This constitutes small deflection theory and as such large deflection surfaces shall not be considered.

1.1.3 Objectives

In order to meet the above aim the following objectives have been established:

i. Identify the differential geometry relations and notation necessary to derive and solve fourth-order partial differential equations.

ii. Derive the governing equations describing biharmonic surfaces under deformation utilising the notation set out above.

iii. Identify the analogue between plate flexure and slow viscous flow of fluids.

iv. Derive a surface discretisation which is C^1 continuous, suitable for the solution of fourth-order differential equations.

v. Develop a new methodology for an arbitrary finite element with a generic configuration of control points.

vi. Express the governing stream function equation using the above discretisation.

vii. Develop an iterative solution methodology, enabling efficient solution of large multi-degree-of-freedom discretisations.

viii. Validate the finite element implementation on simple test cases by exploiting the analogy between Stokes flow and linear plate bending.

ix. Define an approach for imposing both initial and boundary conditions for the stream function surface equation for flow past a circular cylinder.

x. Devise a novel temporally non-linear algorithm enabling the solution of the full Navier-Stokes equations for high Reynolds number flows.

xi. Apply the stream function finite element formulation to the test case of flow past a circular cylinder over range of Reynolds numbers generating complex surface geometries.

xii. Verify the accuracy of the methodology against existing experimental and numerical results.

xiii. Reproduce through numerical simulation the intricate hierarchical geometry of vortices that emerge naturally in the boundary layer of the cylinder.

xiv. Demonstrate the implementation's applicability to generic arbitrary boundary geometries through the simulation of additional test cases.

1.2 Thesis structure

The structure of this thesis takes the following form, satisfying the objectives identified. It comprises of four parts; Research Context, Mathematical Preliminaries, Derivation and Implementation of Theory and the Appendices.

The rest of part one (chapter two) is of a review of previous literature in the subject of surface definition and generation. Following this, the mathematical preliminaries necessary to meet this thesis' aim are set out in part two; chapters three, four and five.

The achievement of the first objective is demonstrated in chapter three. Using this differential geometry notation sections 4.1 and 4.3 of chapter four define the equations for infinitesimal plate deformation and the stream function surface respectively, satisfying objective two. The analogue between these two physically distinct problems is then demonstrated in section 4.6 meeting the third objective

identified. Lastly in part two, chapter five details the existing methodology of subdivision surface discretisation satisfying objective four.

Part three of this thesis then constitutes the original work of this research. Objectives five and six are met in chapter six entitled Formulation of the Numerical Method. A simple linear plate bending formulation is then developed to validate this new subdivision methodology in chapter seven, demonstrating the seventh objective. Three linear benchmarking test cases are then successfully modelled meeting objective eight.

The fully nonlinear implementation is developed in chapter eight establishing objectives nine and ten. This novel implementation is then applied to numerous test cases of flow past a cylinder and validated against existing results in chapter nine. Here objectives eleven and twelve are successfully satisfied over a range of Reynolds numbers. Simulations are performed at high Reynolds number accurately capturing complex vortex formations satisfying objective thirteen, the central goal of the research. Two additional test cases are then implemented in line with the fourteenth objective illustrating the flexibility and potential for the proposed implementation.

Chapter two

Literature review

2.0 Introduction

The subject of this research is the generation of complex geometries describing the formation of vortices in viscous fluid flow. This shall be achieved through solving the fourth-order partial differential equation defining the stream function surface. It will be shown in chapter six that when discretised this requires continuity of the first derivatives. In computer-aided geometric design this is equivalent to a C^1 continuity.

Considerable work towards defining efficient representations for smooth surfaces and complex geometries has being conducted and is discussed below. In particular in recent years research has been in the context of computer animation and modelling for films.

In the case of fluid flow processing power has now reached the stage that simplified forms of the Navier-Stokes equations are being implemented in real time for computer gaming. Stam[10, 11] presents such an approach. Here, rather than being concerned with the mathematical accuracy of the fluid flow, emphasis is with creating convincing imagery through a fast and stable solution. The algorithms are designed to be inherently robust such that the solution never *blows up* or diverges.

In this thesis however we shall be interested in solving the Navier-Stokes equations as accurately as possible, whilst pushing for increasingly complex flow patterns. The equations shall be expressed in a stream function surface form. This formulation leads to a fourth-order partial differential equation, analogous for slow flows to the biharmonic equation governing plate bending. This concept is well known, a number of references[8, 12-15] are given dating back to an article published in the Philosophical Magazine by Lord Rayleigh[16] in 1893. Acknowledging this analogy between the stream function and infinitesimal plate flexure opens up a number of interesting areas for precedence in solving these equations.

Modelling of bending surfaces has seen a good deal of research in the fields of computer animation. Simulation of fabric is an interesting example. As Choi and Ko[17] note, 90 percent of a person's body is typically covered in clothing and as such when striving for realistic character animation the convincing modelling of fabric behaviour is imperative. In all animation importance is placed with visual appearance and with speed. However this naturally means efficient procedures are developed and as such they can be exploited in analogous fields.

In this review reference shall be made to interesting and pertinent areas of research. The emphasis of this thesis is placed on solving these high order differential equations with accuracy. With this aim in mind the documents that appear in this chapter are chosen because of their direct relevance to this thesis' objectives.

2.1 Approaches to surface description and generation

2.1.1 Splines

The central aim of this thesis is to develop an approach for discretising surface geometries. Various approaches have been achieved before and it is useful to first consider the one-dimensional case of the curve. Bézier[18] illustrated that a continuous curve can be derived using a polynomial expression to form weighted averages of a finite number of control points. A similar approach was also developed independently by de Casteljau, (Farin[5] references two earlier unpublished technical reports by de Casteljau[19, 20]). The widely adopted Bézier formation can be seen to be re-expressed using Bernstein[21] polynomials representing the curve in a explicit form enabling direct interrogation[18]. Complex geometries of high continuity can therefore be generated and their properties evaluated. However this Bézier form is based on a single polynomial such that local control of intricate geometries is difficult.

Gordon and Reisenfeld[22], building on the work of de Boor[23], generalised the Bézier methodology to the B-spline. Here a piecewise curve is formed from a limitless composition of polynomial segments. Local control over the curve is possible by manipulating a small number of points in the local region of interest. The zone of influence of the control points and the continuity of the curve is controlled by the order of the B-spline, dictated by the degree of the under lying polynomial. Continuous slope or curvature can thus be achieved as required with localised control.

2.1.2 Interpolatory surfaces

Surfaces can then be formed as a tensor product of two such curves[24, 25]. Such approaches were quickly adopted in the aerospace and automotive industries.

Harder and Desmarais[26] and Meinguet[27] show how the spline was employed for interpolation of scattered data, interpolating arbitrarily positioned evaluation points to form a rational rectangular mesh.

Appa[28] illustrates how this methodology can be applied to interpolate any continuum of data such as temperature, strains *et cetera.* More recently Sarraga[29] and Wang and Wang[30] point to interpolatory approaches for generating a continuum from scattered data by iteratively adjusting the control points of tensor product B-spline surfaces. As Knudson and Nagy[31] describe, a polar based spline interpolation from arbitrarily orientated discrete data values can be implemented where a Cartesian based mesh is not as effective. Here they employ the technique for the generation of geometry for cable net structures.

Using such methods smooth surfaces conforming to a complex irregular array of data points can be achieved. However the underlying structure of the fitting surface is required to be bidirectional.

2.1.3 Bidirectional meshes

The tensor product approach leads to the surface implementation being intrinsically embedded in a bidirectional parameterisation. In general this causes problems where irregular topology meshes are required and the definition of complex models will require multiple surface patches consisting of joining seams. Blending two surfaces together to create a new composite mesh can be problematic using a methodology based around structured meshes. The work of DeRose *et al.*[32] raises this problem of generating complex models with NURBS surfaces through relying on degenerate patches and trimming. As DeRose *et al.* highlight the process of trimming involves high computational expensive whilst only yielding approximate continuity. Stam[33] notes, in some cases at the trimmed interface the surface may not even obtain C^0 continuity.

2.1.4 Chaikin's algorithm

Instead of defining a geometrical problem with an approximate analytical solution[18, 22-25] an alternative approach was taken by de Rham[34] and Chaikin[35]. In these approaches the discretised geometry (the control polygon) was taken as the starting point and simple recursive corner-cutting processes were applied. Chaikin alters the polygon by inserting new points in the ratio of

$\frac{1}{4}$ and $\frac{3}{4}$ along the length of each side. It was shown by Riesenfeld[36] that this approach gives a quadratic B-spline at the limit. This concept of defining an object as the limit of the subdivision process can be translated into higher dimensions.

Peters and Reif[37] illustrate a two dimensional parallel of Chaikin's method where each edge midpoint of a surface control mesh is connected to its corresponding four surrounding edge midpoints. Repeated recursively, this constitutes planing off the angular corners of the polyhedral control mesh until a smooth limit surface is met.

In general however the treatment of a control mesh is more difficult than that of a control polygon and thus rules of greater complexity need to be defined. A polygon, by definition has elements arranged in a linear chain. The connectivity and rules for one-dimensional subdivision are therefore simple. For a control mesh in two dimensions the connectivity can be irregular, therefore topology specific procedures are generally required. These concepts form the basis for surface subdivision.

2.1.5 Subdivision

Loop[38] demonstrated that a single spline curve or polynomial can be *decomposed* into similar splines of half the support. This process can be recursively repeated to the required degree of definition much like Chaikin's algorithm. This theory is given in detail by Loop[38] in his thesis entitled *Smooth subdivision surfaces based on triangles* and also in Zorin and Schröder's[39] *Subdivision for modeling and animation*. Loop applied this methodology to the quartic box spline (see Boehm[40-42]) generalising the relation for arbitrary topology triangular meshes.

Loop[38] showed that through successive application of a constant subdivision matrix that a smooth geometry of continuous tangency, C^1 continuity, can be converged to. Loop's refinement rules were set out for triangular meshes proposing values for the weighting functions to give a surface with continuous

tangent properties. The value of the weighting functions applied is dependent on the valance of the vertex of concern. Here one seamless surface is defined mitigating the need for stitching multiple patches together. Loop's methodology is to be considered in more detail in chapter five.

Similarly an earlier approach by Catmull and Clark[43] was also proposed to counter the problem of arbitrary topology meshes. The approach is based on a recursive B-spline patch affording a rectangular control point mesh. Catmull and Clark define recursive subdivision rules generalised for arbitrary quadrilateral non-rectangular topology meshes C^1 continuity across the surface is given.

2.1.6 Alternative approaches to subdivision

Since the early pioneering work, there have been a number of different standard approaches developed, their differences lying in the topology of mesh they are applicable to and the type of surface they converge to. One key differentiator between the methodologies is their classification in terms of interpolating or approximating. In an approximating scheme the limit surface does not pass thought the initial control points as all the nodal positions are recalculated as weighted averages of the surrounding nodes. Conversely in an interpolatory scheme the original positions of the control vertices are constant and do lie on the limit surface.

Examples of approximating schemes are Loop[38], applicable to arbitrary triangular meshes, and Catmull-Clark[43] and Doo-Sabin[44], supported on rectangular meshes.

Dyn *et al.*[45] and Zorin *et al.*[46] present the Butterfly and Kobbelt[47] and Labisk and Greiner[48] illustrate the $\sqrt{3}$ subdivision scheme, both examples of an interpolatory triangular topology. Precedence for interpolatory rectangular schemes can be found with the $\sqrt{2}$ subdivision of Li *et al.*[49] and the Midedge subdivision schemes as given by Peters and Reif[37] and then by Habib and Warren[50]. Following these approaches Stam and Loop[51] proposed Quad/triangle subdivision a single unified framework based on a hybrid mesh

topology of rectangular and triangle elements. Although as with many discretisation methods this is found to perform badly on meshes which contain elements of bad aspect ratio[51].

Stadler and Holzapfel[52] provide a concise comparison of four of the commonly used methods, Loop, Catmull-Clark, Doo-Sabin and Butterfly, highlighting the significant similarities and differences between the approaches. Ma's *Subdivision surfaces for CAD - an overview*[53] also gives a useful summary on the subject of subdivision. As shown there are obviously a number of different subdivision approaches in development, but as Schröder[54] observes Catmull-Clark and Loop are generally the most commonly employed in practice.

Due to subdivision's inherent basis on polynomials, like splines the conic sections cannot be exactly expressed. To resolve this problem with the spline, as Farin[5] references, the development of a rational form, NURBS curves, was pioneered by Coons[55, 56] and Forrest[57]. Similarly with subdivision approaches to tackle modelling of conic surfaces are being developed. In the work of Jena[58] and Morin *et al.*[59] non-stationary subdivision procedures are demonstrated whereby the subdivision coefficients applied are not constant and are dependent on the level of refinement. With this approach convergence on true conics and surfaces of revolution is possible.

Further extensions of subdivision included non-manifold subdivision of Ying *et al.*[60] and three-dimensional subdivision of Chang *et al.*[61, 62]. These interesting areas of work show that subdivision is an active field of research. However the details of these implementations fall outside of the scope of this thesis.

2.1.7 Parameterisation of arbitrary surfaces

The clear disadvantage of an arbitrary mesh for a two-dimensional surface is the loss of a natural parameterisation. This becomes a problem in applications in which a well parameterised surface is required. As DeRose *et al.*[32] propose a five-dimensional subdivision approach can be employed for texture mapping. Subdivision is not constrained by the dimensionality of space it is applicable in.

To achieve an smooth parameterisation across the surface DeRose *et al.* implement a five-dimensional coordinate consisting of three spatial coordinates and two additional textural parameters (x, y, z, s, t). As DeRose *et al.* explain any potential property of a surface defined as a scalar field f can be determined as a continuous function by assigning values f_i to each mesh vertex across the surface.

In a similar approach extending the subdivision methodology beyond simple two-dimensional surfaces, Weimer and Warren[63] propose a subdivision approach for modelling vector fields. Here they solve systems of linear partial differential equations by defining a coarse set of initial vectors. The process of subdivision can then be applied in areas where a denser solution is required. These approaches[32, 63] could be extended to interpolate any discrete data with continuous first derivative.

2.1.8 Animating bending surfaces

In the animation industries modelling of highly deformable surfaces, fabric being an obvious example, is an area of work which has seen much activity. Ng and Grimsdale[64] give a number of interesting references to early geometrical approaches to representing fabric forms.

Weil[65] developed a technique for modelling draping fabric surfaces from a composition of catenary curves in three-dimensions between fixed suspension points. A surface was then interpolated to approximate these curves achieving a complex surface from a combination of folds.

Following these geometric methods Terzopoulos *et al.*[66] were revolutionary in their approach of using a simplified elasticity theory, numerically solving the discretised differential equations to generate improved realism in their deformed geometry. Further approaches have since been developed, as discussed below, based on a similar finite difference discretisation with increasing complexity.

2.1.9 Finite difference models

In order to model the creases and folds occurring in cloth formations accurately some level of bending stiffness needs to be included. DeRose *et al.*[32] present a mass-spring model to simulate fabric behaviour. Here a rectangular array of *structural springs* is implemented. Shear stiffness is given by triangulated diagonal elements and bending stiffness is achieved though nodal stabilisation springs. The energy functional used for the fabric is *tailored* to the warp and weft direction such that the fabric of the computer animated character's jacket behaves differently depending on whether it is cut on the bias or not[67].

An important issue when modelling fabric behaviour is to capture self-intersection with collision detection. A hierarchical discretisation methodology can be exploited to this achieve this. Bridson *et al.*[68] present an approach using subdivision. The naturally recursive and hierarchical nature of subdivision lends itself to efficiently resolving self-intersection. A similar post-processing hierarchical approach is also implemented by Baraff *et al.*[69] and Choi and Ko[70]. For large deflection extreme distortion and tangling of meshes is an additional problem. Irving *et al.*[71] propose invertible finite elements through extension of their constitutive model to cope with extreme aspect rations and inversion, whilst Palmerio[72] proposes pressure terms as a preventative approach to mesh elements inverting. This problem of collision detection and mesh distortion is naturally a symptom of gross large deflection. This is an issue which is outside the scope of this thesis. Here our implementation is the consideration of a scalar field which can be considered to be an infinitesimal surface deflection.

The initially simplified bending models employed in animation are increasingly sophisticated. Bridson *et al.*[73] and Grinspun *et al.*[74] have shown the expansion from simple mass-spring fabric models to handle more complex bending behaviour including initially curved undeformed geometries. Grinspun *et al.* take care to isolate the two forms of intristic and extrinsic deformation in order to accurately capture the modes of bending. Expanding on the work of Grinspun[75], Burgoon *et al.*[76] apply similar techniques to the modelling of folding and creasing of paper for the simulation of origami models.

As a means of approximation the finite difference method is limited by the error in the truncation of the Taylor series and the coarseness of the grid. Therefore accuracy can be improved by including more terms or including more mesh points. Another limitation is the dependence on a regular grid. Rectangular topology meshes have advantages for describing orthotropic materials and as such they are often used along with their rectangular subdivision counterpart Catmull-Clark[43] for modelling of fabrics[32, 67]. However rectangular meshes can be problematic when modelling complex forms and boundary geometries.

2.1.10 Finite element analysis using rational B-spline curves

Work presented by Hughes *et al.*[77] on integrating computer-aided -design and -engineering show that through using NURBS exact geometrical forms can be reproduced in the analysis model, rather than going through a process of remeshing. This is useful for applications sensitive to geometry such as buckling analysis where imperfections can have a dramatic effect on the accuracy of the result. Unlike the finite difference method, the finite element method can in general accommodate unstructured as well as structured meshes, allowing much greater freedom over the geometry of the problem to be solved. However integrating the finite element approach with B-spline surfaces imbeds in to the methodology the inherent problems that bidirectional parameterised discretisations face with arbitrary meshes and degenerate patches. Methods for coping with irregularity in globally bidirectional meshes can be implemented, such as Fortin and Tanguy's[78] use of hanging nodes. Here linear constraints are written linking the hanging nodes to the global degrees of freedom. Similarly more recently Kagan *et al.*[79, 80] using a B-spline finite element define slave nodes for refinement. This approach goes some way towards defining a unified surface and finite element description. However as Kagan *et al.*[80] identify intervention is necessary to define any local refinement during meshing to obtain full C^1 continuity. Other mesh procedures such as Nikishkov's[81] Quadtree-octree mesh refinement and Hampshire and Topping's[82] extended finite element patches allow irregularity in the mesh. However full C^1 continuity is problematic and it is known that non-conforming C^0 elements naturally give a

reduction in performance especially in the presence of severe element distortion[83, 84].

2.1.11 Subdivision as a finite element

Stam[85, 86] proved for Catmull-Clark[43] and Loop[38] subdivision schemes respectively that properties of the subdivision limit surface can be interrogated without the requirement for actual recursive subdivision. Analytical expressions for the surface were written such that on regular parts of the mesh twelve basis functions could be used to relate the surface properties at any arbitrary parameter value to the original control points.

Following this work Cirak *et al.*[84] first proposed subdivision surfaces as a finite element discretisation. This was based on Loop[38] subdivision exploiting the shape functions as set out by Stam[86]. Utilising the ability to achieve C^1 conformity Cirak *et al.* initially applied the subdivision methodology successfully to rigid thin shell analysis[84]. The subdivision approach enabled efficient discretisation of the governing equations with the required continuity achieved on arbitrary triangular meshes using a simple one-point quadrature numerical integration.

The key differentiator of Cirak *et al.*'s approach is the avoidance of the inadequacy of nodal connections which hampers traditional finite elements and as previously discussed by Irons and Draper[87]. With Cirak *et al.*'s methodology the common interface of two patches exactly corresponds. This gives a single continuous smooth surface as opposed to a collection of non-conforming interpolations. Evaluating a surface using Loop subdivision in this way gives C^2 continuity in areas of regularity and C^1 continuity at the extraordinary valance nodes[39, 88].

Following this success Cirak and Ortiz[89] extended the methodology to model flexible shells in the finite deformation domain. Subdivision is well suited to this latter case. Deriving continuity solely from the nodal deflections eliminates the requirement for rotations to be considered. Finite rotations are more complex to

handle than displacements as the summation of angles requires the multiplication of matrices which is not a commutative process like vector addition[90].

Following Cirak *et al.*[84, 89], Green and Turkiyyah[91, 92] have developed a methodology using a quadrilateral discretisation based on the Catmull-Clark[43] subdivision scheme, again applying their approaches to problems in elasticity.

Dondl *et al.*[93] have also recently adopted a subdivision finite element approach after Cirak and Ortiz[89], applying it to numerically modelling the deformation of thin films.

However to the author's knowledge no work has been carried out applying a subdivision finite element to the solution of the Navier-Stokes equations thus modelling non-linear fluid flow.

2.1.12 Particle based methods

Alternative methods to solving partial differential equations can be used. Choi and Ko[17, 70] propose a non-continuum particle approach to bending surface modelling in contrast to the finite difference or finite element approaches. Here the large deflections required to accurately model buckling are made possible. However their implementation still relies on a structured mesh with each particle having consistent connectivity with the exception of boundaries. Similar connected strands of particles are employed in modelling complex multi-degree of freedom systems such modelling complex hairstyles, as Voino and Magnenat-Thalmann[94] discuss.

Meshless particle methods have been developed, see *An introduction to SPH [Smoothed Particle Hydrodynamics]*, Monaghan[95]. The immediate benefit of a meshless method is with no underlying mesh requiring adaptation, modelling problems of large deformation such as free surface fluid flows and structural fracture are simplified. However the computational costs are in one sense increased as the connectivity must be dynamically updated. Every particle has the

potential to interact with every other. The influence of particles at a distance reduces exponentially through the use of a smoothing kernel. The interactions are therefore greatly simplified by sorting particles into localised regions, considering only interactions in each particles local zone of influence.

Due to the Eulerian nature of fluid flow meshless methods are therefore well suited as a discretisation methodology. Such approaches have been applied to fluids by Monaghan[95] and Kipfer and Westermann[96]. However similar element-free methods are also useful for extreme deformation of structures such as fracture *et cetera* as shown by the dynamic crack propagation work of Belytschko and Tabbara[97].

Put into context of the aim of this thesis, meshless methods require a degree of post-processing in order to generate rational continuum geometries from the discretised data. The benefit of a surface based finite element discretisation is the geometrical surface representation can be naturally expressed using the same basis as the analysis model.

2.1.13 Solution approaches to Navier-Stokes

The governing equations of fluid dynamics, the Navier-Stokes equations, have been defined for centuries[8]. However mathematical solutions exist only for the most simplistic of examples. Therefore until only comparatively recently experiments in fluid dynamics could only be performed physically through the use of flumes and wind tunnels. It is only with the advent of increasingly powerful computers that computational models for interesting test cases are possible.

A large body of work has been executed in the computational solution of the Navier-Stokes equations using a primitive variables form. To list these works is beyond the scope of this thesis. However good references to the subject may be found in references [98-101]. This primitive variables approach leads to a second order differential equation in terms of unknown pressure and velocity, along with the fluid properties, density and viscosity. The requirements on the discretisation

of a second-order differential equation are obviously lower than those imposed on a fourth-order. Only simple geometrical connection between elements, C^0 continuity is necessary. Examples of the solution of this type of problem are abundant[83] and can be achieved on arbitrary topology meshes with no difficulty of the conformity of derivatives at the interface between elements. However this thesis is concerned with solving high order differential equations with stricter requirements on the discretisation methodology.

Formulating the two-dimensional Navier-Stokes equations in a stream function surface form reduces the number of unknowns from three (pressure and two components of velocity) to a single equation in one unknown; the stream function ψ. This approach, although reducing the number of unknowns, increases the order of the problem resulting in the fourth-order partial differential equation of interest in this thesis. Full derivation of this governing equation is given in chapter four.

The fourth derivative places a strict condition on the continuity of any discretisation. An alternative method, sidestepping the fourth-order derivative, is to express the equations in terms of vorticity. Alfonsi and Giorgini[102] and Lesnic *et al.*[103] use such a stream-function-vorticity method. Here one obtains two Poisson's equations instead of the single biharmonic. However this method has disadvantages in that the application of the boundary conditions is not as immediately intuitive. The *no-slip* condition imposes $v^1 = v^2 = 0$ on the velocities at the boundary of the solid. Thus in the stream function formulation the boundary conditions will be with respect to the first derivatives. To visualise this no-slip condition the structural analogy can be used, corresponding to a clamped boundary condition in plate bending. Further discussion of the boundary conditions with regard to their application is given in later chapters.

Little research has been directed into solving the stream function equation, and to the author's knowledge no work has been carried out solving the fourth-order stream function on an arbitrary mesh. This is largely due to the inadequacy of discretisation methods and the restriction of computational power. Exploiting

recent developments in analogous fields means solutions to the stream function surface are now achievable.

2.1.14 Arbitrary meshes and adaptation

As Drake and Manoranjan[104] demonstrate, dynamic mesh adaption can be of great benefit when applied to problems which exhibit areas of localised interest within a larger domain. Multi-scale problems are computationally expensive and as such irregular meshes and adaptive procedures can greatly improve the efficiency. Wansophark and Dechaumphi[105] give an example of finite element mesh refinement applied to computational fluid dynamics an area where efficiency is of great importance. Here the Navier-Stokes equations are in primitive variables form such that a linear element can be used and irregular meshes can be formed with ease and with little constraint.

Work has been carried out on subdivision refinement by Krysl *et al.*[106] and also Müller and Havemann[107] referring to it as *tessellation-on-the-fly.* Amresh *et al*[108] propose methods of refining subdivision meshes based on the surface flatness. The ability to cope with arbitrary meshes with extraordinary valance nodes means that subdivision avoids problems such a hanging nodes that other methods have to contend with. Extensive work is also being carried out in the area of decimation. Mesh simplification through localised removal of vertices and edge collapsing requires by definition an arbitrary mesh. See the work of Schroeder *et al.*[109] and Vieira *et al.*[110]

2.1.15 Unified computer-aided design and engineering

Having developed separately, as Cirak *et al.*[111] state, little work has been done on integrating the two processes of surface modelling and analysis. The works of Soni[112] and Kagan *et al.*[79, 80] are notable in the use of spline based computer-aided geometric design methods for numerical grid generation, bridging the gap between the disciplines. However again these methods encounter the short comings of a bidirectional parameterisation.

Cirak *et al.*[111] present a subdivision approach affording truly integrated modelling as the geometrical surface representation and the finite element are based on the same exactly identical representational framework leading to a single smooth seamless surface.

2.2 Summary

Subdivision as an efficient approach to smooth surface definition has been identified. Precedence for the method's successful application to the discretisation and solution of partial differential equations in the field of elasticity has been shown. Also subdivision as a topological recursive refinement method has been applied to linear slow flow of fluids, where the problem is expressed as a linear partial differential equation.

This thesis develops the subdivision finite element methodologies used in elasticity to enable the modelling of full non-linear, non-steady solutions. The problem of non-linear fluid flow was chosen as it provides an interesting challenge, generating incredibly complex and intricate geometries from elementary initial conditions and boundary conditions. The Navier-Stokes equations are to be expressed in the fourth-order stream function form. As Meleshko[113] says, forms of the biharmonic equation provide a useful test case for the development of numerical methodologies due to the high constraints placed on the discretisation. The subdivision finite element is therefore an ideal candidate for solving this multi-scale problem, ensuring sufficient C^1 continuity on arbitrary meshes. These characteristics enable an irregular mesh to be utilised, efficiently refined in the areas of interest.

In parts two and three of this thesis a novel fully non-linear stream function subdivision finite element is be developed thus generating complex surface geometries describing vortex motion in viscous fluids.

Part two

Mathematical preliminaries

Chapter three

Differential geometry

3.0 Introduction

The purpose of this chapter is to briefly set out the differential geometry required in order to derive and solve the differential equations of interest in this thesis. A curvilinear tensor notation is used enabling efficient description of the equations in an invariant form useful for the derivation of the curvilinear finite element in chapter six.

Differential geometry is the study of curved lines traced over curved surfaces. In general these paths are called metrics and the surfaces are called manifolds. This thesis will be mainly focusing on two-dimensional surfaces embedded in three-dimensional Euclidean space. However the same theory can be extended to describe curved n-dimensional manifolds immersed in a larger dimensioned flat space.

Different situations demand the use of different coordinate systems. Modelling the behaviour of a square plate would naturally be conveniently described with a Cartesian system, a circular plate with a polar coordinate system. In order to describe complex behaviour of generic arbitrary surface geometries it is useful to work in an invariant form, enabling local properties of the manifold to be defined independent of a frame of reference.

Considering any geometry undergoing deformation as topologically invariant, we can define it with respect to a system of curvilinear coordinates. Thompson's *On growth and form*[114] provides a menagerie of examples demonstrating this principle by transforming similar forms in nature, as shown by the process of conformal mapping in Figure 3.1. The simplicity of a general curvilinear form is the equations are written to be applicable in any coordinate system.

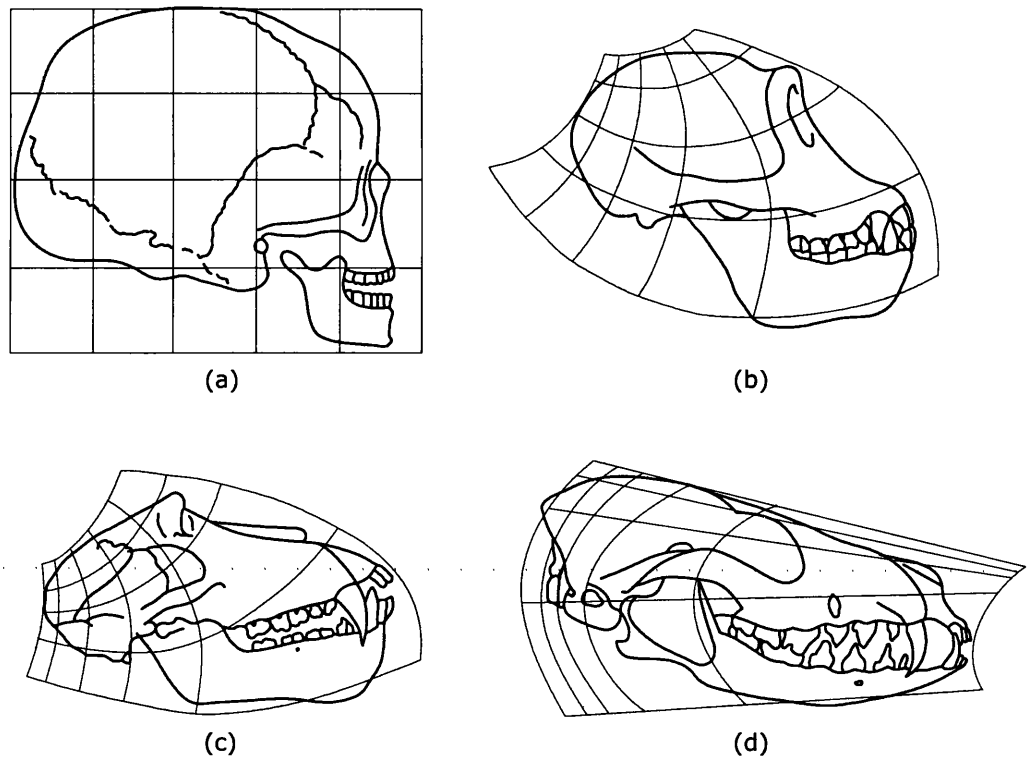


Figure 3.1 Projection of a human skull (a) to a chimpanzee (b) baboon (c) and dog (d) using curvilinear coordinates. Images reproduced from pages 318, 319 and 322 of D'Arcy Thompson's *On growth and form*[114]

3.1 Curvilinear coordinates and tensor notation

When describing the behaviour of materials one is concerned with quantities such as force and stress. Where force is a vector, stress is considered a tensor. Specifically, we can say a scalar is a tensor of rank zero, a vector is a tensor of rank one, and the term tensor when used as above with regard to stress, refers to a tensor of rank two. Tensors of higher orders can also be envisaged. Another fundamental example of a second order tensor is curvature, thus the tensor notation is a powerful tool in combining the theory of surfaces with the theory of continuum mechanics. The following chapter therefore seeks to lay down the basic equations of differential geometry in curvilinear tensor notation ready for use with regard to elasticity and fluid mechanics throughout the following chapters of this dissertation.

With few exceptions, the standard conventions of differential geometry and tensor notation as used by Green and Zerna[115] will be adhered to throughout this text. Hence unless explicitly stated otherwise, the summation convention will be used, where a repeated suffix implies summation for all possible values of that index. In addition, Latin indices take the values 1, 2 or 3 and Greek indices take only the values 1 or 2. For example,

$$\begin{aligned} A^i B_i &= \sum_{i=1}^{3} A^i B_i = A^1 B_1 + A^2 B_2 + A^3 B_3, \\ A^{\alpha}_{.\alpha} &= \sum_{\alpha=1}^{2} A^{\alpha}_{.\alpha} = A^1_{.1} + A^2_{.2}. \end{aligned} \qquad [3.1]$$

The dot before the subscript α in the second equation in [3.1] shows the order of the indices, hence the subscript is second after the superscript. Partial differentiation is donated with a comma, thus the subscripts ,1 and ,2 mean partial differentiation with respect to the coordinates θ^1 and θ^2 respectively, as

$$A_{,i} = \frac{\partial A}{\partial \theta^i}.$$

In order to express our geometry in an invariant form let us have curvilinear coordinates θ^1, θ^2 and θ^3, defined by some transformation from the Cartesian coordinates x^1, x^2 and x^3

$$\theta^i = \theta^i\left(x^1, x^2, x^3\right).$$

In the Cartesian system the vectors $\mathbf{i}_k$ form a unit basis and have the values

$$\mathbf{i}_1 = \mathbf{i}^1 = (1,0,0); \quad \mathbf{i}_2 = \mathbf{i}^2 = (0,1,0); \quad \mathbf{i}_3 = \mathbf{i}^3 = (0,0,1).$$

In general a local basis is formed by the vectors $\mathbf{g}_i$ and $\mathbf{g}^i$, denoted the covariant and contravariant base vectors respectively. This is illustrated in Figure 3.2. The vectors $\mathbf{g}_i$ are directed tangentially along the coordinate curves θ^i, where $\mathbf{g}^1$, $\mathbf{g}^2$ and $\mathbf{g}^3$ are orientated perpendicular to the planes of $\mathbf{g}_2\mathbf{g}_3$, $\mathbf{g}_3\mathbf{g}_1$, and $\mathbf{g}_1\mathbf{g}_2$ respectively, as seen below.

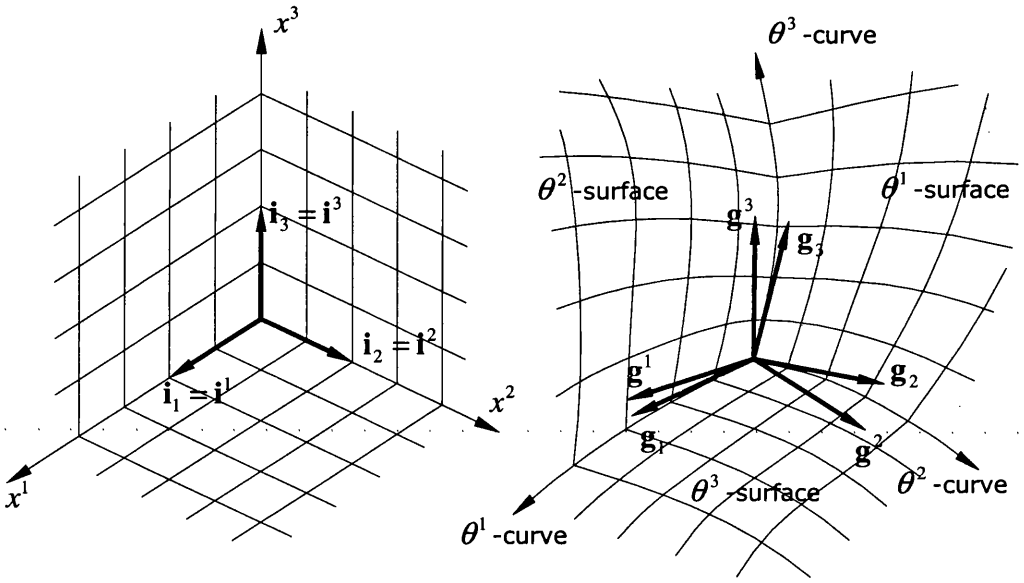


Figure 3.2 Cartesian and curvilinear three-space and their respective covariant and contravariant base vectors, based on Figure 1.5 from Green and Zerna[115] page 20.

The difference between the covariant and contravariant components of a tensor is seen through the rules to which they conform during transformation of coordinates. Passing from one coordinate system θ^{α} to another $\theta^{\hat{\alpha}}$ we have

$$A_{\hat{\alpha}} = \frac{\partial \theta^{\beta}}{\partial \theta^{\hat{\alpha}}} A_{\beta},$$
$$A^{\hat{\alpha}} = \frac{\partial \theta^{\hat{\alpha}}}{\partial \theta^{\beta}} A^{\beta},$$

for the covariant and contravariant components respectively. Fung[116] (on page 38) gives the following examples of each. The differential dx^i transforms as the contravariant components of a vector

$$d\theta^j = \frac{\partial \theta^j}{\partial x^i} dx^i,$$

and the components of the gradient of a scalar ψ, written $\frac{\partial \psi}{\partial x^i}$, conform to the rules of covariant transformation as shown below

$$\frac{\partial \psi}{\partial \theta^j} = \frac{\partial \psi}{\partial x^i} \frac{\partial x^i}{\partial \theta^j}.$$

Similar rules also apply to mixed tensors of any order,

$$A^{\hat{\alpha}}_{.\hat{\beta}} = \frac{\partial \theta^{\hat{\alpha}}}{\partial \theta^{\mu}} \frac{\partial \theta^{\nu}}{\partial \theta^{\hat{\beta}}} A^{\mu}_{.\nu}.$$

Using the above transformation rules we can define our new base as a transformation from the Cartesian

$$\mathbf{g}_j = \frac{\partial x^k}{\partial \theta^j} \mathbf{i}_k, \quad \mathbf{g}^j = \frac{\partial \theta^j}{\partial x^k} \mathbf{i}^k.$$

All the familiar operations performed in a Cartesian system can be written using a curvilinear tensor notation. This is illustrated through notable identities in Table 3.1.

arbitrary vector $\mathbf{v}$:	$\mathbf{v} = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k},$ $= v^1\mathbf{g}_1 + v^2\mathbf{g}_2 + v^3\mathbf{g}_3,$ $= v^i\mathbf{g}_i,$ $= v_i\mathbf{g}^i.$
scalar product:	$\mathbf{u}\cdot\mathbf{v} = u_x v_x + u_y v_y + u_z v_z,$ $= u^i v_i,$ $= u^i v^j g_{ij}.$
vector product:	$\mathbf{u}\times\mathbf{v} = (u_y v_z - u_z v_y)\mathbf{i} +$ $(u_z v_x - u_x v_z)\mathbf{j} +$ $(u_x v_y - u_y v_x)\mathbf{k},$ $= \varepsilon_{ijk} u^i v^j \mathbf{g}^k.$
arbitrary tensor $\mathbf{T}$:	$\mathbf{T} = T_{xx}\mathbf{ii} + T_{xy}\mathbf{ij} + T_{xz}\mathbf{ik} +$ $T_{yx}\mathbf{ji} + T_{yy}\mathbf{jj} + T_{yz}\mathbf{jk} +$ $T_{zx}\mathbf{ki} + T_{zy}\mathbf{kj} + T_{zz}\mathbf{kk},$ $= T^{ij}\mathbf{g}_i\mathbf{g}_j,$ $= T_{ij}\mathbf{g}^i\mathbf{g}^j.$
gradient of a scalar ϕ:	$\nabla\phi = \frac{\partial\phi}{\partial x}\mathbf{i} + \frac{\partial\phi}{\partial y}\mathbf{j} + \frac{\partial\phi}{\partial z}\mathbf{k},$ $= \phi_{,i}\mathbf{g}^i.$

Table 3.1 Useful operations and identities in symbolic, traditional Cartesian and curvilinear tensor notation

gradient of a vector $\mathbf{v}$:

$$\begin{aligned}\nabla\mathbf{v} &= \frac{\partial v_x}{\partial x}\mathbf{ii} + \frac{\partial v_y}{\partial x}\mathbf{ij} + \frac{\partial v_z}{\partial x}\mathbf{ik} + \\ &\quad \frac{\partial v_x}{\partial y}\mathbf{ji} + \frac{\partial v_y}{\partial y}\mathbf{jj} + \frac{\partial v_z}{\partial y}\mathbf{jk} + \\ &\quad \frac{\partial v_x}{\partial z}\mathbf{ki} + \frac{\partial v_y}{\partial z}\mathbf{kj} + \frac{\partial v_z}{\partial z}\mathbf{kk}, \\ &= \mathbf{g}^i \mathbf{v}_{,i}, \\ &= \nabla_i v^j \mathbf{g}^i \mathbf{g}_j.\end{aligned}$$

divergence of a vector $\mathbf{v}$:

$$\begin{aligned}\nabla \cdot \mathbf{v} &= \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}, \\ &= \nabla_i v^i.\end{aligned}$$

Laplacian of a scalar ϕ:

$$\begin{aligned}\nabla^2\phi = \nabla \cdot \nabla\phi &= \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2}, \\ &= g^{ij}\nabla_{ij}\phi, \\ &= \nabla_i^i\phi.\end{aligned}$$

curl of a vector $\mathbf{v}$:

$$\begin{aligned}\nabla\times\mathbf{v} &= \left(\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z}\right)\mathbf{i} + \\ &\quad \left(\frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x}\right)\mathbf{j} + \\ &\quad \left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}\right)\mathbf{k}, \\ &= \varepsilon^{ijk}\nabla_i v_j \mathbf{g}_k.\end{aligned}$$

Table 3.1 Continued.

The above identities illustrate the brevity of the tensor notation. Their relevance is shown considering the case of a two-dimensional surface defined in three-dimensional space.

3.2 Analytic representation of a surface

A typical point in three-dimensional Euclidean space can be defined by the position vector $\mathbf{x}$ with Cartesian coordinates x^k;

$$\mathbf{x} = x^k \mathbf{i}_k .$$

A two-dimensional surface can therefore be defined by expressing its Cartesian coordinates as functions of two parameters θ^α. Referring to Figure 3.2, $\theta_3 = 0$ defines a surface. Thus we have

$$\mathbf{x} = \mathbf{x}\left(\theta^1, \theta^2\right).$$

The convention adopted for two-dimensional work is to denote the base vectors by $\mathbf{a}_i$ and $\mathbf{a}^i$, rather than the $\mathbf{g}$ conventionally used in three-dimensions. Thus the covariant base vectors for a surface are defined as

$$\mathbf{a}_1 = \mathbf{x}_{,1},\ \mathbf{a}_2 = \mathbf{x}_{,2} \text{ and } \mathbf{a}_3 = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\left|\mathbf{a}_1 \times \mathbf{a}_2\right|}, \qquad [3.2]$$

whence $\mathbf{a}_3$ is the surface normal unit vector. This is illustrated in Figure 3.3.

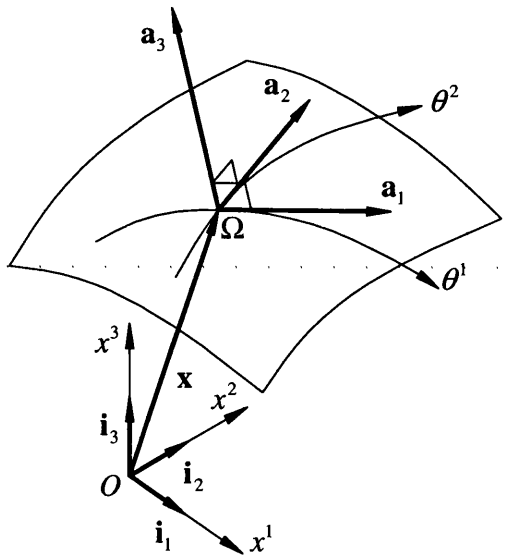


Figure 3.3 Position vector and associated covariant base vectors

The contravariant base vectors are defined by the following relations

$$\mathbf{a}^i \cdot \mathbf{a}_j = \delta^i_j,$$

where the symbols δ^i_j are called the Kronecker deltas, with the following properties:

$$\delta_{ij} = \delta^{ij} = \delta^i_j = 0 \quad (i \neq j),$$
$$\delta_{ij} = \delta^{ij} = \delta^i_j = 1 \quad (i = j, \ \ j \text{ is not summed}).$$

Note the surface normal $\mathbf{a}_3 = \mathbf{a}^3$.

3.3 Fundamental forms of the surface

As Calladine[117] explains in detail, properties of a surface can be considered as either intrinsic or extrinsic. Intrinsic properties are two-dimensional, and thus depend only on lengths measured on the surface. Extrinsic properties however are dependant on the three-dimensional spatial shape of the surface. Thus the inherent behaviour of a surface is, simultaneously, two- and three-dimensional.

3.3.1 Intrinsic properties of a surface

Intrinsic properties of a surface remain invariant under isometric distortion[118]. If a change in geometry is isometric it means that the two states of the surface, before and after deformation, can be mapped one-to-one whilst also preserving all distances. As a simple analogy, lines drawn on a sheet of paper remain the same length even after the paper has been rolled up into a tube or cone.

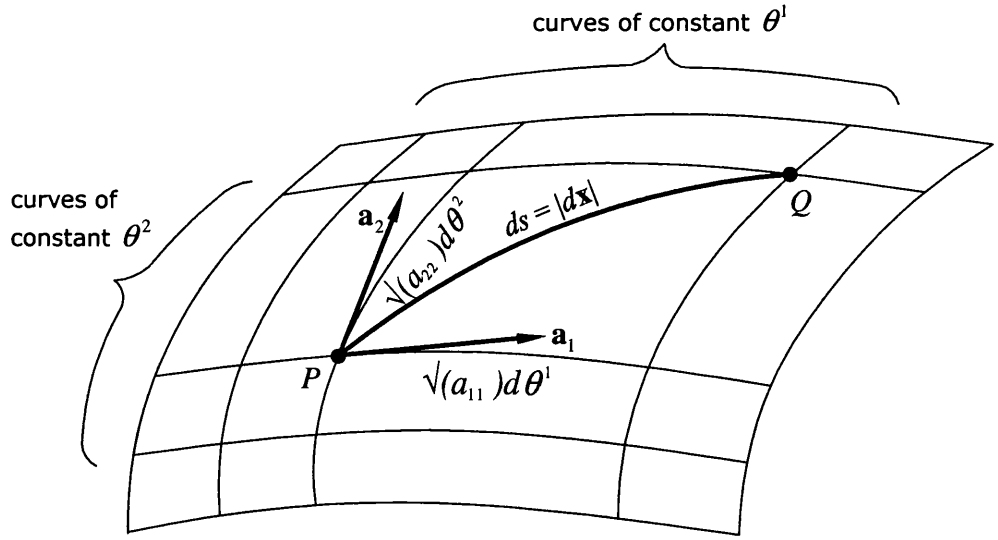


Figure 3.4 Metric between points P and Q on a surface

Figure 3.4 shows an infinitesimal patch of a surface. Adjacent points P and Q are defined by the position vectors $\mathbf{x}$ and $(\mathbf{x}+d\mathbf{x})$, with coordinates θ^α and $\theta^\alpha + d\theta^\alpha$ respectively. The distance ds, as illustrated in the diagram, denotes the magnitude of the vector $d\mathbf{x}$.

From the definition of the covariant base vectors in equation [3.2]:

$$\begin{aligned} d\mathbf{x} &= \mathbf{a}_\alpha d\theta^\alpha , \\ d\mathbf{x}\cdot d\mathbf{x} &= \mathbf{a}_\alpha \cdot \mathbf{a}_\beta d\theta^\alpha d\theta^\beta , \\ ds^2 &= d\mathbf{x}\cdot d\mathbf{x} = a_{\alpha\beta} d\theta^\alpha d\theta^\beta . \end{aligned} \qquad [3.3]$$

where $a_{\alpha\beta}$ are the covariant components of the surface metric tensor defined by the scalar products

$$a_{\alpha\beta} = a_{\beta\alpha} = \mathbf{a}_\alpha \cdot \mathbf{a}_\beta . \qquad [3.4]$$

The corresponding contravariant components are

$$a^{\alpha\beta} = a^{\beta\alpha} = \mathbf{a}^\alpha \cdot \mathbf{a}^\beta . \qquad [3.5]$$

From the definition of the scalar product, the metric tensors must be symmetric, thus the indices are interchangeable.

Equation [3.3] is called the ***first fundamental form*** of the surface. It is the generalisation of the planar Pythagoras' theorem

$$ds^2 = \delta_{ij} dx^i dx^j .$$

The metric tensors can also be defined by the transformations

$$a_{\alpha\beta} = \frac{\partial x^i}{\partial \theta^\alpha} \frac{\partial x^j}{\partial \theta^\beta} \delta_{ij} .$$

In order to perform the vector product we define the permutation symbols, e_{ijk} and e^{ijk}. In three-dimensional Cartesian coordinates we have

$$e_{ijk} = e^{ijk} = \begin{cases} 0 & \text{when } i = j, \ j = k, \text{ or } k = i \\ +1 & \text{when } (i,j,k) \in \{(1,2,3), \ (2,3,1), \ (3,1,2)\} \\ -1 & \text{when } (i,j,k) \in \{(3,2,1), \ (2,1,3), \ (1,3,2)\} \end{cases}$$

They can be used to define the determinant $|a_{ij}|$ as

$$|a_{ij}| = e_{rst} a_{r1} a_{s2} a_{t3} .$$

For our two-dimensional curvilinear case they are denoted $\varepsilon_{\alpha\beta}$, $\varepsilon^{\alpha\beta}$ and are obtained through transformation

$$\begin{aligned} \varepsilon_{\alpha\beta} &= \frac{\partial x^\gamma}{\partial \theta^\alpha} \frac{\partial x^\mu}{\partial \theta^\beta} e_{\gamma\mu}, \\ &= e_{\alpha\beta} \left| \frac{\partial x^\lambda}{\partial \theta^\rho} \right| . \end{aligned}$$

Therefore

$$\begin{aligned}
\varepsilon_{11} &= \varepsilon_{22} = 0, \\
\varepsilon_{12} &= -\varepsilon_{21} = \sqrt{a}, \\
\varepsilon^{11} &= \varepsilon^{22} = 0, \\
\varepsilon^{12} &= -\varepsilon^{21} = \frac{1}{\sqrt{a}}.
\end{aligned}$$

where a is the determinant

$$a = \left| a_{\alpha\beta} \right| = a_{11}a_{22} - (a_{12})^2 .$$

The following relations between the covariant and contravariant base vectors can now be defined through the vector product

$$\begin{aligned}
\mathbf{a}_\alpha \times \mathbf{a}_\beta &= \varepsilon_{\alpha\beta}\mathbf{a}_3, \\
\mathbf{a}^\alpha \times \mathbf{a}^\beta &= \varepsilon^{\alpha\beta}\mathbf{a}_3.
\end{aligned}$$

The lengths of the base vectors $\mathbf{a}_r$ and $\mathbf{a}^r$ are in general not unity. Their magnitudes are

$$\left|\mathbf{a}_i\right| = \sqrt{a_{ii}}, \qquad \left|\mathbf{a}^i\right| = \sqrt{a^{ii}}, \quad (i \text{ not summed}). \tag{3.6}$$

The *physical* components of a vector $\mathbf{v} = v^i\mathbf{a}_i = v_i\mathbf{a}^i$ are therefore

$$v^i\sqrt{a_{ii}}, \qquad v_i\sqrt{a^{ii}}, \quad (i \text{ not summed}). \tag{3.7}$$

These identities can be used to define a small element of area on the θ^i -surface

$$\begin{aligned}
dS_i &= \left|\mathbf{a}_j \times \mathbf{a}_k\right| d\theta^j d\theta^k, \\
&= \left|\varepsilon_{jki}\mathbf{a}^i\right| d\theta^j d\theta^k, \quad (i \text{ not summed}, \ i \neq j \neq k). \\
&= \sqrt{aa^{ii}}\, d\theta^j d\theta^k.
\end{aligned} \tag{3.8}$$

The surface element dS, on the $\theta^3 = 0$ surface, reduces to

$$dS = \sqrt{a}\, d\theta^1 d\theta^2 . \qquad [3.9]$$

3.3.2 Extrinsic properties of a surface

Again considering the point P on a surface, the spatial surface geometry is defined by the curvatures of the surface curves passing through this point. Figure 3.5 illustrates $d\mathbf{a}_3$, the change in unit normal for a small displacement, $d\mathbf{x}$ along such a curve, which can be given as

$$d\mathbf{a}_3 = -b_{\alpha\beta}\mathbf{a}^\alpha d\theta^\beta . \qquad [3.10]$$

where the tensor components $b_{\alpha\beta}$ are given by

$$b_{\alpha\beta} = b_{\beta\alpha} = \mathbf{a}_3 \cdot \mathbf{x}_{,\alpha\beta} = \mathbf{a}_3 \cdot \mathbf{a}_{\alpha,\beta} = -\mathbf{a}_\alpha \cdot \mathbf{a}_{3,\beta} . \qquad [3.11]$$

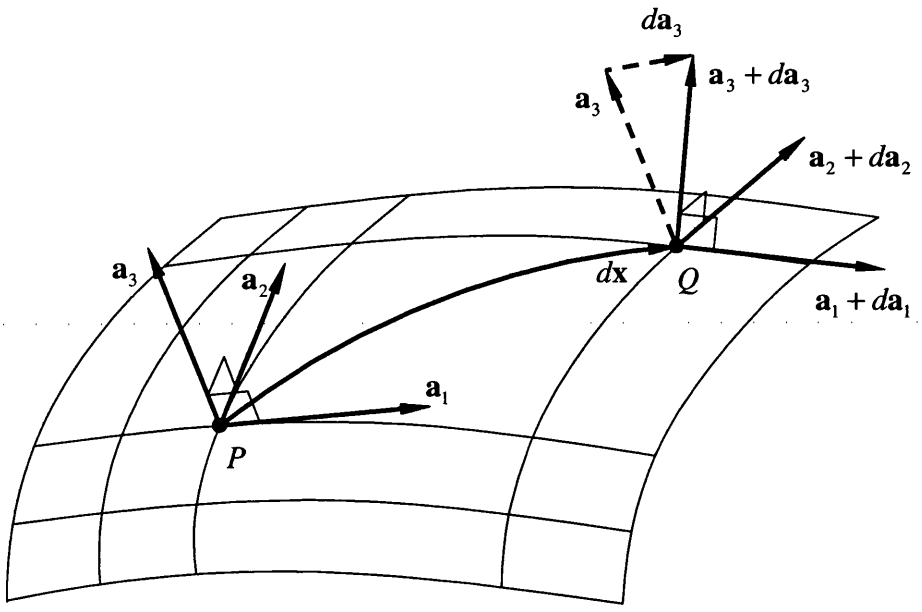


Figure 3.5 Change in the normal vector along a small displacement on a surface

The rate of the change in the surface normal relative to the metric defines the three-dimensional geometry at that point, in that direction. This leads to the ***second fundamental form*** of the surface, defined by the scalar product

$$d\mathbf{x}\cdot d\mathbf{a}_3 = \left(\mathbf{a}_\alpha d\theta^\alpha\right)\cdot\left(-b_{\mu\beta}\mathbf{a}^\mu d\theta^\beta\right) = -b_{\alpha\beta}d\theta^\alpha d\theta^\beta .$$

Therefore the local geometry of a surface can be uniquely defined using two differential expressions, the first and second fundamental forms[119, 120]

$$\begin{aligned} \mathrm{I} &= d\mathbf{x}\cdot d\mathbf{x} = a_{\alpha\beta}d\theta^\alpha d\theta^\beta , \\ \mathrm{II} &= d\mathbf{x}\cdot d\mathbf{a}_3 = -b_{\alpha\beta}d\theta^\alpha d\theta^\beta . \end{aligned} \qquad [3.12]$$

$b_{\alpha\beta}$ are therefore the coefficients of the second fundamental form.

There is a third fundamental form. However this can be expressed in terms of the first and second forms, and thus is less useful.

3.4 Curvature of a surface

Surface curvature at a point is derived from the infinite number of surface curves passing through that point. Taking one such curve, the components of $d\mathbf{a}_3$ parallel to and perpendicular to $d\mathbf{x}$ define the normal and twisting curvature of the surface in the direction of the metric. Thus the normal curvature in the direction of $d\mathbf{x}$, is given by equations [3.12]:

$$\kappa = -\frac{d\mathbf{x}\cdot d\mathbf{a}_3}{d\mathbf{x}\cdot d\mathbf{x}} = -\frac{\mathrm{II}}{\mathrm{I}} .$$

The curvature κ is a function of the orientation of the metric in the plane of the surface, $\kappa(\lambda)$, where $\lambda = d\theta^2/d\theta^1$. Thus curvature is a second order tensor and can be expressed in terms of its two orthogonal principal directions. These are

the directions in which the normal curvature is a maximum and minimum and the twisting curvature is zero.

This condition is satisfied when $d\mathbf{x}$ and $d\mathbf{a}_3$ are parallel

$$d\mathbf{a}_3 + \kappa\, d\mathbf{x} = 0,$$

which can be rewritten using equations [3.10] and [3.3] as

$$\left(-b_{\mu\beta}\mathbf{a}^{\mu} + \kappa\,\mathbf{a}_{\beta}\right)d\theta^{\beta} = 0.$$

Scalar multiplying by $\mathbf{a}^{\alpha}$ gives

$$-\left(b^{\alpha}_{\beta} - \kappa\,\delta^{\alpha}_{\beta}\right)d\theta^{\beta} = 0, \qquad [3.13]$$

where $b^{\alpha}_{\beta} = a^{\alpha\mu}b_{\mu\beta}$ which is symmetric.

Excluding trivial solutions, equation [3.13] gives a quadratic in κ

$$\left|b^{\alpha}_{\beta} - \kappa\delta^{\alpha}_{\beta}\right| = 0,$$

and

$$\kappa^2 - \kappa\left(b^1_1 + b^2_2\right) + b^1_1 b^2_2 - b^2_1 b^1_2 = 0,$$

with solutions $\kappa_{(1)}$ and $\kappa_{(2)}$, the principal curvatures.

The surface can then be characterised by defining the **mean curvature**, H, as the mean of the principal curvatures

$$\begin{aligned} H &= \frac{\kappa_{(1)} + \kappa_{(2)}}{2} = \frac{b^1_1 + b^2_2}{2} = \frac{b^{\alpha}_{\alpha}}{2}, \\ &= \frac{a_{11}b_{22} - 2a_{12}b_{12} + a_{22}b_{11}}{2\left(a_{11}a_{22} - a_{12}a_{12}\right)}, \end{aligned} \qquad [3.14]$$

and the **Gaussian curvature**, K, the product of the principal curvatures

$$K = \kappa_{(1)}\kappa_{(2)} = b_1^1 b_2^2 - b_1^2 b_2^1 = \frac{\left(b_\alpha^\alpha b_\beta^\beta - b_\alpha^\beta b_\beta^\alpha\right)}{2}, \\ = \frac{b_{11}b_{22} - b_{12}b_{12}}{a_{11}a_{22} - a_{12}a_{12}}. \qquad [3.15]$$

We can then say

$$\kappa = \kappa_{(1)} \cos^2 \alpha + \kappa_{(2)} \sin^2 \alpha,$$

where the angle α describes the orientation of the direction $d\theta^2/d\theta^1$. The above equation is known as Euler's theorem[120]. This concept of expressing a variable in an arbitrary orientation in terms of the principal values is familiar with stress tensors, and like the Mohr's circle for stress, curvature can be illustrated in two dimensions through a circle diagram[121].

3.5 Christoffel symbols and covariant differentiation

As illustrated above, when analysing surfaces we are concerned with tensor fields, thus we need to cope with differentiation. Here we define $\Gamma_{\alpha\beta}^{\mu}$, the Christoffel symbols of the second kind. The Christoffel symbols, together with the coefficients of the second fundamental form, quantify the rotation and expansion of the base vectors as we travel along the coordinate curves[122]. This is seen in the *Gauss equation*[120] written

$$\mathbf{a}_{\alpha,\beta} = \Gamma_{\alpha\beta}^{\mu}\mathbf{a}_\mu + b_{\alpha\beta}\mathbf{a}_3. \qquad [3.16]$$

Scalar multiplying the Gauss equation [3.16] by $\mathbf{a}^\gamma$ defines the Christoffel symbols of the second kind as

$$\Gamma_{\alpha\beta}^{\gamma} = \mathbf{a}^\gamma \cdot \mathbf{a}_{\alpha,\beta}. \qquad [3.17]$$

Defining a surface vector $\mathbf{v} = v^{\alpha}\mathbf{a}_{\alpha} = v_{\alpha}\mathbf{a}^{\alpha}$, the partial derivative is

$$\begin{aligned} \mathbf{v}_{,\beta} &= v^{\alpha}{}_{,\beta}\mathbf{a}_{\alpha} + v^{\alpha}\mathbf{a}_{\alpha,\beta}, \\ &= v_{\alpha,\beta}\mathbf{a}^{\alpha} + v_{\alpha}\mathbf{a}^{\alpha}{}_{,\beta}. \end{aligned} \qquad [3.18]$$

The covariant derivative is the partial derivative along the coordinate curves with the additional correction terms which tell how the coordinates have changed. Adopting the notation of ∇_{α} for covariant differentiation, we have

$$\begin{aligned} \nabla_{\beta} v^{\alpha} &= v^{\alpha}{}_{,\beta} + \Gamma^{\alpha}_{\mu\beta} v^{\mu}, \\ \nabla_{\beta} v_{\alpha} &= v_{\alpha,\beta} - \Gamma^{\mu}_{\alpha\beta} v_{\mu}, \end{aligned} \qquad [3.19]$$

and thus, with the Gauss equation [3.16], the partial derivative [3.18] can be written as

$$\begin{aligned} \mathbf{v}_{,\beta} &= \nabla_{\beta} v^{\alpha}\mathbf{a}_{\alpha} + v^{\alpha} b_{\alpha\beta}\mathbf{a}_3, \\ &= \nabla_{\beta} v_{\alpha}\mathbf{a}^{\alpha} + v^{\alpha} b_{\alpha\beta}\mathbf{a}_3. \end{aligned} \qquad [3.20]$$

3.6 Summary

This chapter has defined the necessary differential geometry relations using a curvilinear tensor notation successfully achieving the first objective identified in §1.1.3.

These results shall be utilised throughout this thesis in the derivation and the solution of the fourth-order differential equations. Table 3.2 over the page provides a concise summary of the pertinent relations and conventions for reference.

covariant base vectors $\mathbf{a}_i$:

$$\mathbf{a}_1 = \mathbf{x}_{,1},\ \mathbf{a}_2 = \mathbf{x}_{,2} \text{ and } \mathbf{a}_3 = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{|\mathbf{a}_1 \times \mathbf{a}_2|} \quad [3.2]$$

components of the metric tensor:

$$a_{\alpha\beta} = a_{\beta\alpha} = \mathbf{a}_\alpha \cdot \mathbf{a}_\beta \quad [3.4]$$

$$a^{\alpha\beta} = a^{\beta\alpha} = \mathbf{a}^\alpha \cdot \mathbf{a}^\beta \quad [3.5]$$

coefficients of the second fundamental form $b_{\alpha\beta}$:

$$b_{\alpha\beta} = b_{\beta\alpha} = \mathbf{a}_3 \cdot \mathbf{x}_{,\alpha\beta}. \quad [3.11]$$

lengths of the base vectors:

$$|\mathbf{a}_i| = \sqrt{a_{ii}}, \quad |\mathbf{a}^i| = \sqrt{a^{ii}}, \quad [3.6]$$

(i not summed).

physical components of a vector:

$$v^i\sqrt{a_{ii}}, \quad v_i\sqrt{a^{ii}}, \quad [3.7]$$

(i not summed).

surface element dS_i:

$$dS_i = \sqrt{aa^{ii}}\, d\theta^j d\theta^k, \quad [3.8]$$

(i not summed, $i \neq j \neq k$).

surface element dS on the $\theta^3 = 0$ surface:

$$dS_3 = dS = \sqrt{a}\ d\theta^1 d\theta^2, \quad [3.9]$$

Christoffel symbols of the second kind:

$$\Gamma^\gamma_{\alpha\beta} = \mathbf{a}^\gamma \cdot \mathbf{a}_{\alpha,\beta}. \quad [3.17]$$

covariant derivative ∇_α:

$$\nabla_\beta v_\alpha = v_{\alpha,\beta} - \Gamma^\mu_{\alpha\beta} v_\mu, \quad [3.19]$$

partial derivative of $\mathbf{v}$:

$$\mathbf{v}_{,\beta} = \nabla_\beta v_\alpha \mathbf{a}^\alpha + v^\alpha b_{\alpha\beta}\mathbf{a}_3. \quad [3.20]$$

Table 3.2 Summary of the differential geometry relations given in chapter three

Chapter four

Continuum mechanics for surface definition

4.0 Introduction

The purpose of this chapter is to derive the governing fourth-order differential equations to be solved in this thesis. Here we are interested in two formulations. The first half of this chapter, §4.1-4.2, analyses the problem of infinitesimal elasticity theory applied to bending of a thin isotropic plate. It is shown that this problem satisfies the solution of the biharmonic equation. The second half of this chapter, §4.3-4.4, formulates the Navier-Stokes equations of viscous fluid motion in the fourth-order stream function form.

Comparison of these two governing equations realises the analogue between the flexure of plates and slow viscous flow of fluids. This shall be exploited in chapter seven enabling validation of this thesis' proposed implementation on linear test cases before extending the model to full non-linear fluid behaviour.

4.1 Elasticity for plate flexure

4.1.1 Geometry of surface structures under deformation

The form of a *thin* shell-like surface structure can be defined fully by offsetting a single surface, generating a volume bounded by the two new similar exteriors. Therefore, assuming a thickness t, we can define such an object using a middle surface of parameters θ^{α}, and the parameter θ^3 restricted to the interval

$$-\tfrac{1}{2}t \le \theta^3 \le \tfrac{1}{2}t$$

Similar to the equations given in chapter three, we can obtain the following expression for the shell position vector

$$\mathbf{r} = \mathbf{r}\left(\theta^1, \theta^2, \theta^3, x^0\right) = \mathbf{x}\left(\theta^1, \theta^2, x^0\right) + \theta^3 \mathbf{a}_3\left(\theta^1, \theta^2, x^0\right). \qquad [4.1]$$

Here, in this section, x^0 is introduced denoting time, avoiding confusion with the thickness t. This relationship in equation [4.1] is illustrated in Figure 4.1.

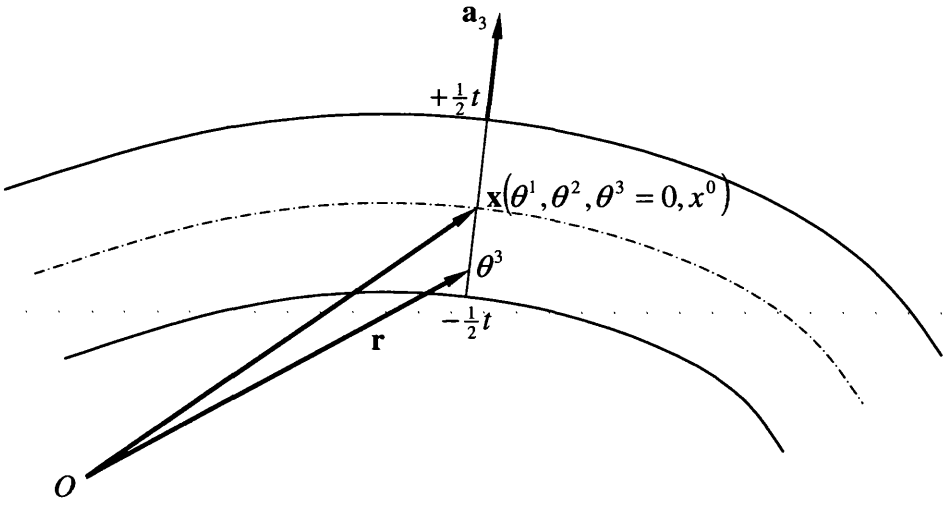


Figure 4.1 Shell position vector for deformed configuration

The convention of upper-case characters denoting the undeformed reference geometry and lower-case as the current configuration is adhered to as adopted by Zienkiewicz *et al.*[123]

In classical elasticity theory the description of a body's deformation depends upon the deviation of the current state from the body's initial unstrained configuration. Commonly the initial state is known and thus the position vector of the body is expressed as a function of this known reference.

This formulation is known as *Lagrangian*, referring to the *material* coordinate axes, tracking the path of particles through deformation

$$x^i = x^i\left(X^1, X^2, X^3, x^0\right).$$

Conversely one can work in what is known as *Eulerian* form with reference to the *spatial* axes, considering a fixed point in space

$$X^i = X^i\left(x^1, x^2, x^3, x^0\right).$$

In addition to the material and spatial coordinate systems it is useful to set up a system of curvilinear coordinates of the middle surface of the shell, to be convected with the continuum as it deforms. Figure 4.2 illustrates a shell-like structure undergoing deformation to a new strained configuration at time x^0 from an initial configuration at time $x^0 = 0$ in relation to this set of bases. For simplicity the two coordinate systems $OX_1X_2X_3$ and $ox_1x_2x_3$ are superimposed. However in general as Mase[124] describes, the material and spatial coordinates may be defined with a relative translation and rotation.

For the majority of this chapter we shall be concerned with infinitesimal deformations and thus this issue of convected versus Lagrangian or Eulerian coordinates becomes irrelevant as these descriptions are equivalent. However

this will not be the case during the discussion of fluid flow. Here the subtle difference between Lagrangian and Eulerian coordinates will be revisited.

In general before limiting ourselves to small deformation theory, we can define the shell position vectors in the deformed and undeformed states as

$$\begin{aligned}\mathbf{r}(\theta^1,\theta^2,\theta^3,x^0)&=\mathbf{x}(\theta^1,\theta^2,x^0)+\theta^3\mathbf{a}_3(\theta^1,\theta^2,x^0),\ \text{with}-\frac{t}{2}\le\theta^3\le\frac{t}{2},\\ \mathbf{R}(\theta^1,\theta^2,\theta^3)&=\mathbf{X}(\theta^1,\theta^2)+\theta^3\mathbf{A}_3(\theta^1,\theta^2),\ \text{with}-\frac{T}{2}\le\theta^3\le\frac{T}{2}.\end{aligned} \qquad [4.2]$$

The base vectors can all be referred to $\mathbf{X}$ and $\mathbf{x}$

$$\begin{aligned}\mathbf{a}_\alpha&=\mathbf{x}_{,\alpha},\\ \mathbf{a}_3&=\mathbf{g}_3=\mathbf{r}_{,3},\\ \mathbf{g}_\alpha&=\mathbf{r}_{,\alpha}=\mathbf{a}_\alpha+\theta^3\mathbf{a}_{3,\alpha},\\ \mathbf{A}_\alpha&=\mathbf{X}_{,\alpha},\\ \mathbf{A}_3&=\mathbf{G}_3=\mathbf{R}_{,3},\\ \mathbf{G}_\alpha&=\mathbf{R}_{,\alpha}=\mathbf{A}_\alpha+\theta^3\mathbf{A}_{3,\alpha}.\end{aligned} \qquad [4.3]$$

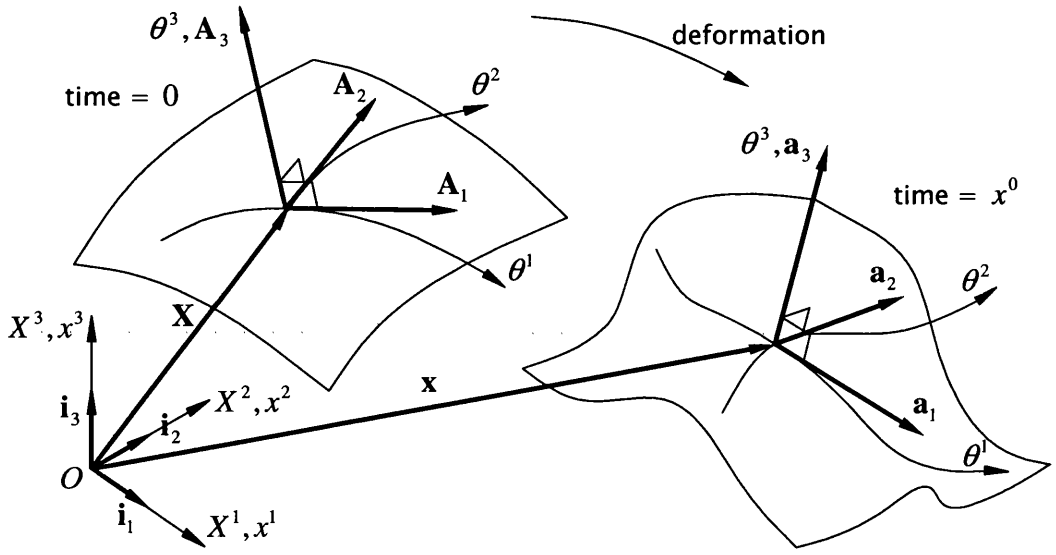


Figure 4.2 Shell surface in reference and current configurations with associated Cartesian and convected surface bases

Finally the covariant and contravariant components of the metric tensors in their deformed and undeformed states can therefore be defined as

$$\begin{aligned} a_{\alpha\beta} &= \mathbf{a}_\alpha \cdot \mathbf{a}_\beta, & a^{\alpha\beta} &= \mathbf{a}^\alpha \cdot \mathbf{a}^\beta, \\ g_{ij} &= \mathbf{g}_i \cdot \mathbf{g}_j, & g^{ij} &= \mathbf{g}^i \cdot \mathbf{g}^j, \\ A_{\alpha\beta} &= \mathbf{A}_\alpha \cdot \mathbf{A}_\beta, & A^{\alpha\beta} &= \mathbf{A}^\alpha \cdot \mathbf{A}^\beta, \\ G_{ij} &= \mathbf{G}_i \cdot \mathbf{G}_j, & G^{ij} &= \mathbf{G}^i \cdot \mathbf{G}^j. \end{aligned} \qquad [4.4]$$

4.1.2 Infinitesimal elasticity: Plate theory

The case of interest here is that of a flat plate spanning horizontally whilst being subjected to a lateral load, as illustrated in Figure 4.3. In small deflection theory the transverse deflection ψ is a scalar through the assumption of

$$\frac{\partial \psi}{\partial \theta^1} \text{ and } \frac{\partial \psi}{\partial \theta^2} << 1. \qquad [4.5]$$

This relation is such that the bending and stretching can be assumed to be independent of one another. This is a consequence of the fact that a body infinitesimally thin in one dimension can undergo a curvature change whilst metrics on the surface remain unaffected[66].

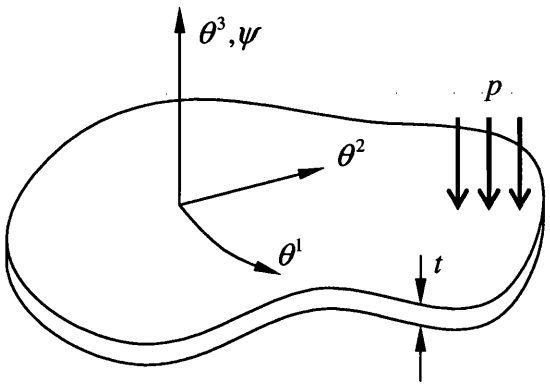


Figure 4.3 Flat plate with lateral loading p

Therefore for pure bending we can say

$$\frac{A}{a} = 1. \qquad [4.6]$$

In addition plane sections taken normal to the plate surface are not sheared and remain plane, the so called Kirchoff hypothesis. Finally, for completeness we assume a homogenous, isotropic material at constant temperature.

4.1.3 Strain-displacement relations

The deformation gradient $\mathbf{F}$ defined by

$$\mathbf{F} = \frac{\partial \mathbf{r}}{\partial \mathbf{R}} \text{ or } F_{ij} = \frac{\partial r_i}{\partial R_j},$$

is the fundamental description of the body's displacement. However this includes rigid body rotation information. Therefore as a measure of *actual strain* the difference between the position vectors is needed[125]. A requirement of any strain measurement is that it must be insensitive to rigid body motion. Here we define the Green-Lagrange strain tensor which measures the difference of the square of the line elements

$$d\mathbf{r} \cdot d\mathbf{r} - d\mathbf{R} \cdot d\mathbf{R} = (dr)^2 - (dR)^2. \qquad [4.7]$$

From equation [4.4] we have the definitions

$$(dr)^2 = g_{ij} d\theta^i d\theta^j,$$
$$(dR)^2 = G_{ij} d\theta^i d\theta^j.$$

Thus the strain tensor satisfies the equation

$$(dr)^2 - (dR)^2 = 2\gamma_{ij} d\theta^i d\theta^j, \qquad [4.8]$$

where

$$\gamma_{ij} = \tfrac{1}{2}(g_{ij} - G_{ij}).$$

Using the definitions written in equation [4.3], neglecting higher orders in θ^3, we can rewrite the strain in the form

$$\begin{aligned}
\gamma_{ij} &= \tfrac{1}{2}(g_{ij} - G_{ij}), \\
&= \tfrac{1}{2}(\mathbf{g}_i \cdot \mathbf{g}_j - \mathbf{G}_i \cdot \mathbf{G}_j), \\
&= \tfrac{1}{2}\left[(\mathbf{a}_i \cdot \mathbf{a}_j + 2\theta^3 \mathbf{a}_\alpha \cdot \mathbf{a}_{3,\alpha} + \theta^3\theta^3 \mathbf{a}_{3,\alpha} \cdot \mathbf{a}_{3,\alpha})\right. \\
&\quad \left. -(\mathbf{A}_i \cdot \mathbf{A}_j + 2\theta^3 \mathbf{A}_\alpha \cdot \mathbf{A}_{3,\alpha} + \theta^3\theta^3 \mathbf{A}_{3,\alpha} \cdot \mathbf{A}_{3,\alpha})\right], \\
&= \alpha_{ij} + \theta^3 \beta_{ij}.
\end{aligned}$$

The tensors α_{ij} and β_{ij} constitute the stretching and bending strains respectively†. The unstrained geometry is given as planar and imposing the condition of equation [4.6] we have

$$\begin{aligned}
&\alpha_{ij} \to 0, \text{ as } \frac{\mathbf{a}_i}{\mathbf{A}_i} \to 1 \\
&\beta_{\alpha\beta} = -b_{\alpha\beta} \text{ as } \mathbf{A}_{\alpha,\beta} \cdot \mathbf{A}_3 = 0
\end{aligned} \qquad [4.9]$$

Now from the definitions given in chapter two our coefficients of the second fundamental equation of the surface can be given as

$$b_{\alpha\beta} = \nabla_{\alpha\beta} \psi.$$

---

† For a one-dimensional body, a line element, the overall strain tensor would be made up of three parts, with the addition of a twist component.

and the strain tensor reduces simply to

$$\gamma_{\alpha\beta} = \theta^3 \nabla_{\alpha\beta} \psi \,. \qquad [4.10]$$

4.1.4 Equations of stress

The stress tensor τ^{ij} illustrated in Figure 4.4, associated with the covariant base vectors, describes the state of stress in the body. For thin plates the assumption of plane stress is made, as τ^{i3}, τ^{3i} are zero on the plate surfaces and may be deemed negligible within the depth.

A small element of plate is bounded by the surfaces θ^α = constant, $\theta^\alpha + d\theta^\alpha$ = constant and $\theta^3 = \pm\frac{1}{2}t$. Taking one side it is assumed there is a distribution of stress throughout the depth of the plate such that the midplane is unstrained.

Recalling from the previous chapter a small element of surface is given by

$$dS_\alpha = \sqrt{a a^{\alpha\alpha}}\, d\theta^\beta d\theta^3, \quad (\alpha \neq \beta).$$

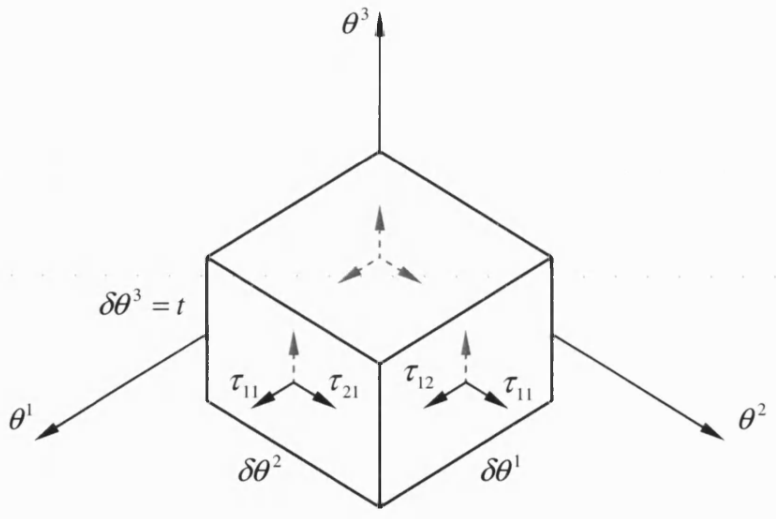


Figure 4.4 Stresses acting a small element of material

The physical component of force acting on an element of θ^α-surface as a result of this stress is therefore

$$\mathbf{T}_\alpha \sqrt{a^{\alpha\alpha}}\, d\theta^\beta d\theta^3 = \tau^{\alpha\lambda}\mathbf{a}_\lambda \sqrt{aa^{\alpha\alpha}}\, d\theta^\beta d\theta^3, \quad (\alpha \neq \beta),$$

which simplifies to

$$\mathbf{T}_\alpha = \sqrt{a}\tau^{\alpha\lambda}\mathbf{a}_\lambda . \qquad [4.11]$$

The physical stress couples per unit length as a result of these forces are given by

$$\mathbf{m}_\alpha = \frac{1}{\sqrt{aa^{\alpha\alpha}}} \int_{-\frac{t}{2}}^{\frac{t}{2}} (\mathbf{a}_3 \times \mathbf{T}_\alpha)\theta_3 d\theta_3 , \qquad [4.12]$$

where the line element along the surface intersection of $\theta^3 = 0$ and $\theta^\beta =$ constant is,

$$\sqrt{a_{\alpha\alpha}}\, d\theta^\alpha = \sqrt{aa^{\beta\beta}}\, d\theta^\alpha, \text{ (indices not summed, } \alpha \neq \beta).$$

Then substituting equation [4.11] into [4.12], the stress couples $\mathbf{m}_\alpha$ can be related directly to the stress tensors $\tau^{\alpha\beta}$, by

$$\mathbf{m}_\alpha \sqrt{a^{\alpha\alpha}} = m^{\alpha\rho}\mathbf{a}_3 \times \mathbf{a}_\rho ,$$

where

$$m^{\alpha\rho} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \tau^{\alpha\rho}\theta_3 d\theta_3 .$$

Using the permutation tensors we can relate $\mathbf{m}_\alpha$ to the contravariant vectors by

$$\mathbf{m}_\alpha \sqrt{a^{\alpha\alpha}} = \varepsilon_{\rho\lambda} m^{\alpha\rho} \mathbf{a}^\lambda . \qquad [4.13]$$

The physical components of $\mathbf{m}_\alpha$ are given in terms of the unit base vectors along the θ^α = constant coordinate curves,

$$\mathbf{m}_\alpha = m_{(\alpha 1)} \frac{\mathbf{a}^2}{\sqrt{a^{22}}} + m_{(\alpha 2)} \frac{\mathbf{a}^1}{\sqrt{a^{11}}} .$$

Using the convention of Green and Zerna[115] in which the brackets indicate these are physical components and not tensors. These physical components are graphically represented in Figure 4.5.

$m_{(11)}$ and $m_{(22)}$ can be envisaged as the *bending* moments of the plate, whereas $m_{(12)} = -m_{(21)}$ can be attributed to the *twisting* action of the plate.

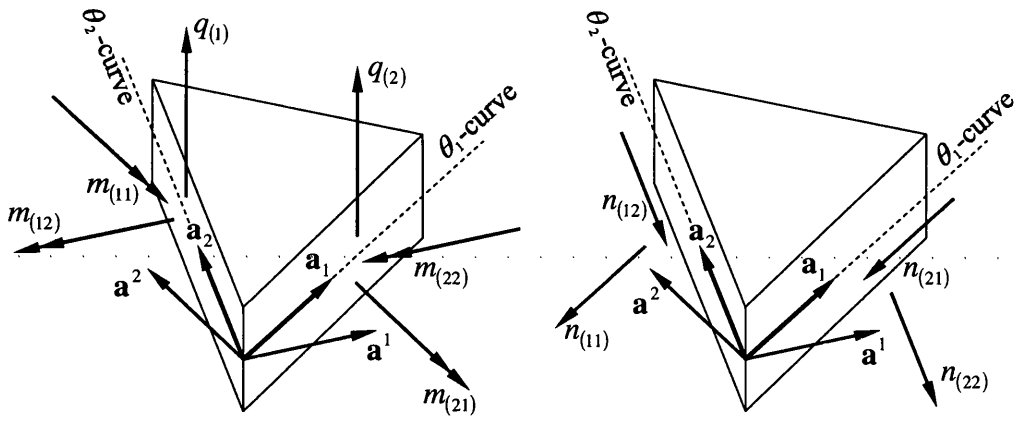


Figure 4.5 Physical stress couples and resultants generated through bending and stretching of a small element of plate (where the double headed arrows represent the axis about which the moment is acting conforming to the right hand rule)

In addition to stress couples, the transverse shear components of the stress tensor generate stress resultants $\mathbf{n}_\alpha$. Under the restrictions of this example there are no membrane forces lying in the plane of the plate. Thus the stress resultants are given simply as

$$\mathbf{n}_\alpha \sqrt{a^{\alpha\alpha}} = q^\alpha \mathbf{a}_3 ,$$

where

$$q^\alpha = \int_{-\frac{t}{2}}^{\frac{t}{2}} \tau^{\alpha 3} d\theta_3 .$$

Imposing moment equilibrium about the θ^μ-axis, whilst neglecting lower order terms in $\nabla_\alpha q^\alpha$ and p,

$$\left[\frac{\partial}{\partial\theta^\alpha}\left(\frac{\varepsilon_{\rho\lambda}}{\sqrt{a^{\alpha\alpha}}} m^{\alpha\rho}\mathbf{a}^\lambda\right)\sqrt{aa^{\alpha\alpha}}\theta^1 d\theta^2\right]\cdot\mathbf{a}_\mu -$$
$$\left[\left(\frac{1}{\sqrt{a^{\alpha\alpha}}} q^\alpha \mathbf{a}_3 \times \mathbf{a}_\alpha\right)\sqrt{aa^{\alpha\alpha}}\, d\theta^1 d\theta^2\right]\cdot\mathbf{a}_\mu = 0,$$

$$\sqrt{a}\varepsilon_{\rho\lambda}\nabla_\alpha m^{\alpha\rho}\mathbf{a}^\lambda\cdot\mathbf{a}_\mu - \sqrt{a}\varepsilon_{\alpha\beta}q^\alpha\mathbf{a}^\beta\cdot\mathbf{a}_\mu = 0, \qquad [4.14]$$

which simplifies to give

$$\nabla_\alpha m^{\alpha\rho} - q^\rho = 0. \qquad [4.15]$$

Imposing vertical equilibrium gives,

$$\nabla_\alpha q^\alpha + p = 0. \qquad [4.16]$$

Differentiating equation [4.15] and substituting into [4.16] gives the single equation,

$$\nabla_{\alpha\beta} m^{\alpha\beta} + p = 0. \qquad [4.17]$$

4.1.5 Constitutive equations

For a body which at time $x^0 = 0$ is unstressed we can relate the state of stress τ^{ij} to the strain γ_{rs} through the relation

$$\tau^{ij} = E^{ijrs}\gamma_{rs}, \qquad [4.18]$$

where E^{ijrs} are the contravariant elastic coefficients depending on the metric tensor and physical properties of the body. These 81 terms are simplified through symmetry,

$$E^{ijrs} = E^{jirs} = E^{ijsr} = E^{rsij},$$

thus there are potentially 21 elastic coefficients. However for the considered case of an isotropic material, symmetry further reduces this count to just two independent coefficients. As introduced above when discussing stress, for our purposes of plate bending we can restrict our interests to two dimensions. Thus in order to be directionally independent, that is isotropic, the primary stresses and strains are aligned and therefore we must have a relationship in symbolic form something like

$$\boldsymbol{\tau} = A\boldsymbol{\gamma} + B\mathbf{I}tr(\boldsymbol{\gamma}),$$

or in component form

$$\tau^{\alpha\beta} = A\gamma^{\alpha\beta} + Ba^{\alpha\beta}a^{\lambda\mu}\gamma_{\lambda\mu}.$$

This, we know, can be written in terms of Young's modulus of elasticity E and Poisson's ratio ν, giving

$$\tau^{\alpha\beta} = \frac{E}{1-\nu^2}\left[(1-\nu)a^{\alpha\rho}a^{\beta\lambda} + \nu a^{\alpha\beta}a^{\rho\lambda}\right]\gamma_{\rho\lambda}. \qquad [4.19]$$

In traditional Cartesian notation this agrees with our plane stress relations, as given by Timoshenko and Goodier[126],

$$\sigma_x = \frac{E}{1-\nu^2}\left(\varepsilon_x + \nu\varepsilon_y\right),$$
$$\sigma_y = \frac{E}{1-\nu^2}\left(\varepsilon_y + \nu\varepsilon_x\right),$$
$$\tau_{xy} = \frac{E}{1+\nu}\gamma_{xy},$$

and conversely the relations

$$\varepsilon_x = \frac{1}{E}\left(\sigma_x - \nu\sigma_y\right),$$
$$\varepsilon_y = \frac{1}{E}\left(\sigma_y - \nu\sigma_x\right),$$
$$\gamma_{xy} = \frac{1+\nu}{E}\tau_{xy},$$

where γ_{xy} is the mathematical strain, equal to one half of the engineering strain.

4.1.6 Formulation of the governing equation

Combining equations [4.10] and [4.19] we obtain the relations between the stress tensor and the curvature of the plate

$$\tau^{\alpha\beta} = \frac{E\theta^3}{1-\nu^2}\left[(1-\nu)\nabla^{\alpha\beta}\psi + \nu a^{\alpha\beta}\nabla^{\lambda}_{\lambda}\psi\right]. \qquad [4.20]$$

Multiplying by θ^3 and integrating over the depth of the plate gives

$$\int_{-\frac{t}{2}}^{\frac{t}{2}} \theta^3 \tau^{\alpha\beta} d\theta^3 = \frac{E}{1-\nu^2}\left[(1-\nu)\nabla^{\alpha\beta}\psi + \nu a^{\alpha\beta}\nabla^{\lambda}_{\lambda}\psi\right]\int_{-\frac{t}{2}}^{\frac{t}{2}} \left(\theta^3\right)^2 d\theta^3 ,$$

which simplifies to

$$m^{\alpha\beta} = \frac{Et^3}{12\left(1-\nu^2\right)}\left[(1-\nu)\nabla^{\alpha\beta}\psi + \nu a^{\alpha\beta}\nabla^{\lambda}_{\lambda}\psi\right].$$

or

$$m^{\alpha\beta} = D\left[(1-\nu)\nabla^{\alpha\beta}\psi + \nu a^{\alpha\beta}\nabla^{\lambda}_{\lambda}\psi\right], \qquad [4.21]$$

where D is the plate stiffness given by

$$D = \frac{Et^3}{12\left(1-\nu^2\right)}. \qquad [4.22]$$

Substituting equation [4.21] in the expression for the stress couple given in equation [4.17], we have

$$D\left[(1-\nu)\nabla^{\alpha\beta}_{\alpha\beta}\psi + \nu\nabla^{\alpha\lambda}_{\alpha\lambda}\psi\right] + p = 0. \qquad [4.23]$$

Finally the Poisson's ratio ν can be eliminated from the expression (although still appearing in D) obtaining

$$\nabla^{\alpha\beta}_{\alpha\beta}\psi = \frac{p}{D}. \qquad [4.24]$$

4.2 The biharmonic equation for plate bending

For the special case of no load, where $p=0$, we have the plate deformation described by the fourth-order equation

$$\boxed{\nabla^{\alpha\beta}_{\alpha\beta}\psi=0.} \qquad [4.25]$$

This is the ***biharmonic equation*** in general form.

This can be expressed as

$$a^{\alpha\rho}a^{\beta\lambda}\nabla_{\rho\lambda\alpha\beta}\psi=0,$$

and written in full in Cartesian coordinates as

$$\frac{\partial^4\psi}{\partial x^4}+\frac{2\partial^4\psi}{\partial x^2\partial y^2}+\frac{\partial^4\psi}{\partial y^4}=0,$$

or in symbolic form as

$$\nabla^4\psi=0.$$

The biharmonic equation describes the behaviour of a number of physically unrelated phenomena. The case of interest in this dissertation is the viscous flow of fluids. The following section gives a brief explanation of the basic equations of fluids to allow comparison with the infinitesimal elasticity theory above, showing that this problem of creeping flow is also satisfied by the biharmonic equation. For a full description of fluid behaviour the interested reader is directed to comprehensive texts[14, 98, 100, 127].

4.3 Viscous flow of fluids

4.3.1 Eulerian and Lagrangian descriptions

The description of fluid deformation or flow is much more complex than for elastic solids. In structures we are interested in defining the geometry relative to the initial state. However, as White[8] describes, with fluids we are more concerned with the velocity field, rather than defining the actual position of particles. Excepting extreme nonlinear situations, the arrangement of particles in a structure remains largely ordered. In fluids the opposite is true, thus a Eulerian description is more useful. However the conservation laws describing the behaviour of continua used below are Lagrangian, referring to masses of particles. Thus we need to provide a link between our two descriptions.

We shall consider only two-dimensional flow such that, in Eulerian Cartesian notation we have the following definition for the velocity vector $\mathbf{v}$ at a given point

$$\begin{aligned}\mathbf{v} &= \mathbf{v}\left(x^1, x^2, t\right), \\ &= \mathbf{i}_\alpha v^\alpha\left(x^1, x^2, t\right),\end{aligned}$$

and $v_3 = 0$.

The rate of change of a quantity ϕ at a particular point in the fluid domain is given by the Eulerian local derivative

$$\frac{\partial \phi}{\partial t}.$$

To express this in Lagrangian terms, referring to a specific fluid *particle*, one must take into account the effects of advection. Thus the Lagrangian, particle derivative is written as

$$\begin{aligned}\frac{D\phi}{Dt} &= \frac{\partial \phi}{\partial t} + v_x \frac{\partial \phi}{\partial x} + v_y \frac{\partial \phi}{\partial y}, \\ &= \frac{\partial \phi}{\partial t} + v^\alpha \phi_{,\alpha}.\end{aligned}$$

Applying the Lagrangian derivative to the velocity we have the following expression for acceleration

$$\frac{Dv^{\alpha}}{Dt} = \frac{\partial v^{\alpha}}{\partial t} + v^{\beta}\nabla_{\beta}v^{\alpha}, \qquad [4.26]$$

in Cartesian notation

$$\frac{Dv_x}{Dt} = \frac{\partial v_x}{\partial t} + v_x\frac{\partial v_x}{\partial x} + v_y\frac{\partial v_x}{\partial y},$$
$$\frac{Dv_y}{Dt} = \frac{\partial v_y}{\partial t} + v_x\frac{\partial v_y}{\partial x} + v_y\frac{\partial v_y}{\partial y},$$

or in symbolic vector form

$$\frac{D\mathbf{v}}{Dt} = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}.$$

The first term on the right-hand side of equation [4.26] is the *local acceleration* and the second is the *convective acceleration* [128], the latter of which constitutes the nonlinearity of the equations.

4.3.2 Deformation of Newtonian fluids

The spatial derivative of the velocity vector at a point is a tensor and thus, like any arbitrary tensor, it can be decomposed into its symmetric and antisymmetric parts[128], the strain tensor $\boldsymbol{\gamma}$ and the vorticity tensor $\boldsymbol{\omega}$. In symbolic notation these are

$$\boldsymbol{\gamma} = \frac{1}{2}\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^{\mathrm{T}}\right),$$

$$\boldsymbol{\omega} = \frac{1}{2}\left(\nabla\mathbf{v} - (\nabla\mathbf{v})^{\mathrm{T}}\right).$$

These have covariant components

$$\gamma_{\alpha\beta} = \frac{1}{2}\left(\nabla_{\alpha}v_{\beta} + \nabla_{\beta}v_{\alpha}\right), \qquad [4.27]$$

$$\omega_{\alpha\beta} = \frac{1}{2}\left(\nabla_{\alpha}v_{\beta} - \nabla_{\beta}v_{\alpha}\right). \qquad [4.28]$$

In three dimensions vorticity Ω^k is a vector given by the vorticity tensor as

$$\Omega^k = \frac{1}{2}\varepsilon^{ijk}\omega_{ij}.$$

This thesis is using a two-dimensional formulation. The vorticity Ω is a scalar given by the vorticity tensor as

$$\Omega = \frac{1}{2}\varepsilon^{\alpha\beta}\omega_{\alpha\beta}.$$

Figure 4.6 illustrates the distortion of an element of fluid over time δt, comprising of translation, rotation, dilatation and shearing. Vorticity equals the rate of rotation, as shown below, where we can see the rotation of the element at the limit is given by

$$\omega_z = \left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}\right).$$

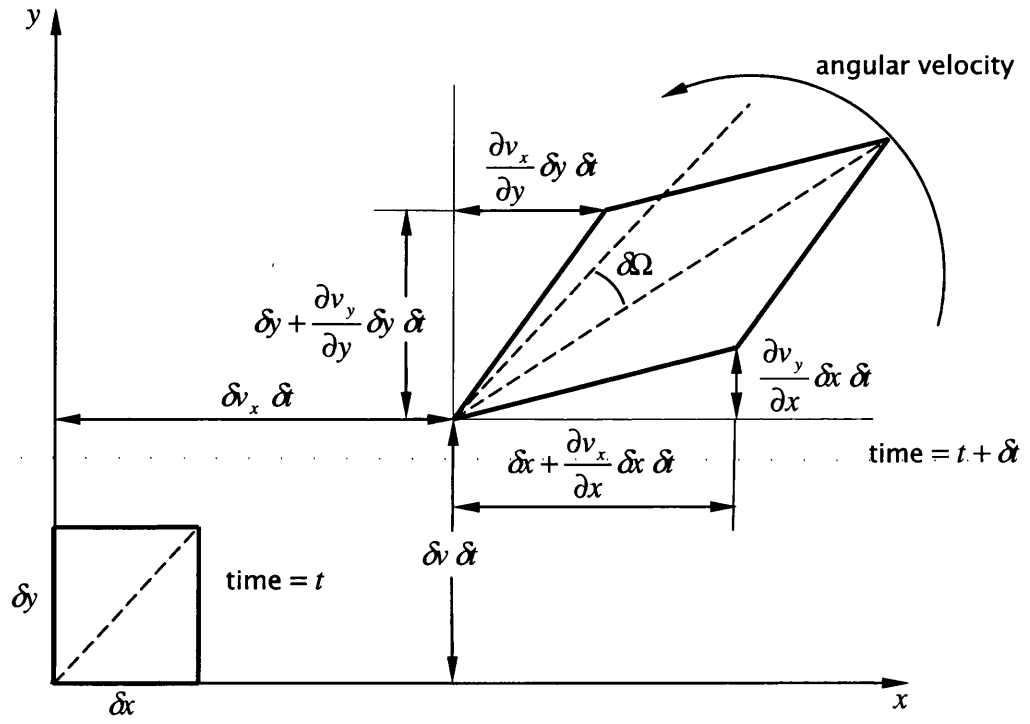


Figure 4.6 An element of fluid undergoing deformation, reproduced from White[8] page 19

4.3.3 Continuity equation

The continuity equation is a statement of conservation of mass. In a Lagrangian form this is simply m = constant. For the considered case of incompressible flow the density of the fluid ρ is also a constant, therefore the conservation of mass implies a conservation of volume and so we can impose the condition that the dilation in the two perpendicular axes must be equal and opposite giving a divergence equal to zero.

In traditional Cartesian form this is

$$\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}=0,$$

or in tensor notation

$$\nabla_\alpha v^\alpha = 0, \qquad [4.29]$$

or in symbolic form

$$\nabla \cdot \mathbf{v} = 0.$$

4.3.4 Momentum equations

Newton's second law of motion relates the rate of change of momentum of a body to the force acting on it

$$F^i = ma^i.$$

Following the useful standard convention, as White[8] and Versteeg and Malalasekera[127], this is written with respect to volumes as opposed to mass, giving

$$\rho\frac{Dv^\alpha}{Dt} = f^\alpha. \qquad [4.30]$$

The acceleration must be written using the Lagrangian particle derivative as we are concerned with a particular element of fluid.

The force on the right hand side is a result of the stress $\tau^{\alpha\beta}$ being applied to a small piece of fluid. Working in two dimensions additional body forces such as those caused by a gravitational field are neglected. We therefore have the force per unit volume

$$f^{\beta} = \nabla_{\alpha}\tau^{\alpha\beta}. \qquad [4.31]$$

Substituting for f^{α}, the momentum equation [4.30] in terms of stress becomes

$$\rho\frac{Dv^{\beta}}{Dt} = \nabla_{\alpha}\tau^{\alpha\beta}. \qquad [4.32]$$

in symbolic form

$$\rho\frac{D\mathbf{v}}{Dt} = \nabla\cdot\boldsymbol{\tau},$$

where the divergence of the tensor $\boldsymbol{\tau}$ results in a vector.

For an isotropic Newtonian fluid $\tau^{\alpha\beta}$ is a linear function of $\gamma^{\alpha\beta}$ as given by Versteeg and Malalasekera[127]

$$\tau^{\alpha\beta} = -pa^{\alpha\beta} + 2\mu\gamma^{\alpha\beta} + a^{\alpha\beta}\lambda\gamma^{\eta}_{\eta},$$

where p is the pressure, μ is the first or dynamic viscosity, to be denoted as simply viscosity hereafter, and λ is the second viscosity. μ relates stresses to strains, whereas λ relates stresses to the changes in volume.

Thus for incompressible flow the above equation simplifies to

$$\tau^{\alpha\beta} = -pa^{\alpha\beta} + 2\mu\gamma^{\alpha\beta}. \qquad [4.33]$$

4.3.5 The Navier-Stokes equations

Substituting equation [4.33] into [4.32], along with [4.29] we obtain the ***incompressible Navier-Stokes equations of fluid motion*** in general curvilinear tensor notation

$$\rho\left(\frac{\partial v^{\alpha}}{\partial t}+v^{\beta}\nabla_{\beta}v^{\alpha}\right)=-a^{\alpha\beta}\nabla_{\beta}p+\mu a^{\beta\gamma}\nabla_{\beta\gamma}v^{\alpha},$$
$$\nabla_{\alpha}v^{\alpha}=0. \qquad [4.34]$$

Alternatively in two-dimensional component form these are given by

$$\rho\left(\frac{\partial v_x}{\partial t}+v_x\frac{\partial v_x}{\partial x}+v_y\frac{\partial v_x}{\partial y}\right)=-\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 v_x}{\partial x^2}+\frac{\partial^2 v_x}{\partial y^2}\right),$$
$$\rho\left(\frac{\partial v_y}{\partial t}+v_x\frac{\partial v_y}{\partial x}+v_y\frac{\partial v_y}{\partial y}\right)=-\frac{\partial p}{\partial y}+\mu\left(\frac{\partial^2 v_y}{\partial x^2}+\frac{\partial^2 v_y}{\partial y^2}\right). \qquad [4.35]$$
$$\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}=0.$$

or in symbolic form,

$$\rho\frac{D\mathbf{v}}{Dt}=-\nabla p+\mu\nabla^2\mathbf{v},$$
$$\nabla\cdot\mathbf{v}=0.$$

We therefore have three equations in three unknowns - the pressure and two components of velocity. These equations are highly nonlinear such that solutions to them are extremely complex. Analytic solutions do not exist for all but the simplest conditions, thus numerical solution procedures have been developed.

Solving the equations in the form of unknown velocities and pressure, as in equation [4.35] is called a *primitive variables* approach. Iterative guess-and-

correct solution procedures such as the SIMPLE† algorithm[127] exist to tackle the solving of these coupled equations. In this dissertation the above equations are reformulated and solved using the stream function ψ, providing the link with the elasticity theory. This approach in one sense simplifies the problem, but it is also shown that in doing this, additional more restrictive constraints are imposed on the solution.

4.3.6 The stream function

Introducing the stream function ψ, the condition for incompressible flow is satisfied by

$$v_x = \frac{\partial \psi}{\partial y},$$
$$v_y = -\frac{\partial \psi}{\partial x}.$$

In tensor notation exploiting the permutation symbol these are

$$v^\alpha = \varepsilon^{\alpha\beta}\nabla_\beta \psi. \qquad [4.36]$$

Therefore contour lines on the stream function surface, lines of constant ψ, are *streamlines* and are tangent to the flow, with the mass flux across them equal to zero.

Using these relations eliminates the continuity equation leaving us with just the two equations of motion. Substituting equation [4.36] into [4.34] gives

$$\rho\left(\frac{\partial}{\partial t}\left(\varepsilon^{\alpha\beta}\nabla_\beta\psi\right)+\varepsilon^{\gamma\eta}\nabla_\eta\psi\nabla_\gamma\left(\varepsilon^{\alpha\beta}\nabla_\beta\psi\right)\right)=-a^{\alpha\beta}\nabla_\beta p+\mu a^{\gamma\pi}\nabla_{\gamma\pi}\left(\varepsilon^{\alpha\beta}\nabla_\beta\psi\right),$$

† The acronym standing for Semi-Implicit Method for Pressure Linked Equation.

which can be written

$$\varepsilon^{\alpha\beta}\left(\frac{\partial}{\partial t}+\varepsilon^{\gamma\eta}\nabla_\eta\psi\nabla_\gamma-\frac{\mu}{\rho}a^{\gamma\eta}\nabla_{\gamma\eta}\right)\left(\nabla_\beta\psi\right)=-a^{\alpha\beta}\nabla_\beta p\,,$$

or

$$a_{\alpha\chi}\varepsilon^{\chi\beta}\left(\frac{\partial}{\partial t}+\varepsilon^{\gamma\eta}\nabla_\eta\psi\nabla_\gamma-\frac{\mu}{\rho}a^{\gamma\eta}\nabla_{\gamma\eta}\right)\left(\nabla_\beta\psi\right)=-\nabla_\alpha p\,.$$

Finally the unknown pressure p can be eliminated through cross differentiation,

$$\varepsilon^{\alpha\pi}a_{\alpha\chi}\varepsilon^{\chi\beta}\nabla_\pi\left(\frac{\partial}{\partial t}+\varepsilon^{\gamma\eta}\nabla_\eta\psi\nabla_\gamma-\frac{\mu}{\rho}a^{\gamma\eta}\nabla_{\gamma\eta}\right)\left(\nabla_\beta\psi\right)=-\varepsilon^{\alpha\lambda}\nabla_{\alpha\lambda}p=0\,,$$

and therefore writing

$$\begin{aligned}\varepsilon^{\alpha\pi}a_{\alpha\chi}\varepsilon^{\chi\beta}\nabla_\pi &= a^{\pi\lambda}\varepsilon_{\chi\lambda}\varepsilon^{\chi\beta}\nabla_\pi,\\ &= a^{\pi\lambda}\delta^\beta_\lambda\nabla_\pi,\\ &= a^{\pi\beta}\nabla_\pi,\end{aligned}$$

we obtain

$$\left(\frac{\partial}{\partial t}+\varepsilon^{\gamma\eta}\nabla_\eta\psi\nabla_\gamma-\frac{\mu}{\rho}a^{\gamma\eta}\nabla_{\gamma\eta}\right)\left(a^{\alpha\beta}\nabla_{\alpha\beta}\psi\right)=0\,. \qquad [4.37]$$

4.4 The stream function form of the Navier-Stokes equations

This Navier-Stokes equations are now expressed in a single equation in only one variable, the stream function ψ

$$\boxed{\left(\frac{\partial}{\partial t}+\varepsilon^{\gamma\eta}\nabla_\eta\psi\nabla_\gamma-\frac{\mu}{\rho}a^{\gamma\eta}\nabla_{\gamma\eta}\right)\left(a^{\alpha\beta}\nabla_{\alpha\beta}\psi\right)=0\,.} \qquad [4.37]$$

The penalty for this simplification is the increase in order of the differentials. The equation we need to solve is now a forth-order partial differential equation, greatly increasing the requirements on the continuity of the solution. Previously the Navier-Stokes equations in primitive variables formulation contained only second order differentials.

The stream function equation [4.37] can be written in traditional Cartesian notation

$$\frac{\partial}{\partial t}\left(\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}\right)+\frac{\partial\psi}{\partial y}\frac{\partial}{\partial x}\left(\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}\right)-\frac{\partial\psi}{\partial x}\frac{\partial}{\partial y}\left(\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}\right)$$
$$-\frac{\mu}{\rho}\left(\frac{\partial^4\psi}{\partial x^4}+\frac{2\partial^4\psi}{\partial x^2\partial y^2}+\frac{\partial^4\psi}{\partial y^4}\right)=0,$$

and in symbolic form utilising the del operator ∇

$$\frac{\partial}{\partial t}\left(\nabla^2\psi\right)+\left(\nabla\left(\nabla^2\psi\right)\right)\times\left(\nabla\psi\right)=\frac{\mu}{\rho}\left(\nabla^4\psi\right).$$

4.5 Reynolds number

In engineering experimentation dimensional analysis is used to make comparisons between models, physical, theoretical and computational.† In fluid dynamics the nondimensional Reynolds number Re is used to ensure similitude.

$$Re=\frac{\rho VL}{\mu}, \qquad [4.38]$$

where V and L are the characteristic velocity and length respectively. ρ is the density and μ is the viscosity of the fluid.

† Galileo Galilei was known to highlight the significance of relative dimensions as Thompson[114] quotes: *"a dog could probably carry two or three such dogs on his back; but I believe a horse could not even carry one of his own size"*

4.6 Plate bending stream function analogy

Comparison of equations [4.25] and [4.37] reveals the analogy between plate flexure and viscous flow of fluids. For very low Reynolds numbers, when $Re << 1$, corresponding to viscous dominated flow, the inertia terms can be neglected. Applying this condition to equation [4.37], one sees this *creeping flow*, also named Stokes flow, satisfies the biharmonic equation, exactly identical to equation [4.25]

$$\nabla^{\alpha\beta}_{\alpha\beta}\psi = 0$$

or

$$\frac{\partial^4\psi}{\partial x^4} + \frac{2\partial^4\psi}{\partial x^2\partial y^2} + \frac{\partial^4\psi}{\partial y^4} = 0,$$

or

$$\nabla^4\psi = 0.$$

Interestingly, and fundamental to the theory of viscous fluids, if conversely the Reynolds number is very large the inertia terms do not dominate everywhere. As White[8] remarks, in the boundary layer the viscous terms are *never* negligible. This is a result of the *no-slip condition* which states that at a boundary with a solid object, the velocity must be zero[100]. This subsequently causes the $\nabla^4\psi$ term to be very large in the boundary layer. Thus for modelling of fluids interacting with bodies the viscous terms are of particular interest.

4.7 Conclusions

The fourth-order partial differential equations of interest in this thesis have successfully been derived utilising the curvilinear tensor notation detailed in chapter three. The two equations to be descretised and solved in the third part of this thesis are the biharmonic equation of plate bending, equation [4.25], and

the fourth-order stream function equation [4.37]. The comparison of these equations has also led to the analogy between viscous flow of fluids and plate bending being realised demonstrating the third objective identified in chapter one.

Chapter five

Subdivision surface discretisation

5.0 Introduction

An objective of this thesis is to derive a surface discretisation approach which enables interrogation of the continuum's second derivative and thus demanding continuity of the first derivate. It will be shown in part three that the governing fourth-order equations derived in chapter four, when discretised, require this level of C^1 continuity.

In this chapter subdivision as a surface discretisation methodology is introduced. An existing approach proposed by Loop[38] based on an arbitrary triangular topology mesh whilst meeting the above requirements is set out.

5.1 Subdivision surfaces

The principle of subdivision is that at the limit of recursive application of simple refinement rules to an initially coarse control mesh a smooth geometry can be converged on. In this thesis Loop subdivision[38], first proposed by Charles Loop in 1987is used.

The premise behind Loop subdivision is to recursively subdivide each triangle in the control mesh into four similar triangles by inserting new vertices on the midpoint of each edge of the face. This routine is illustrated in Figure 5.1 using a linear weighting of the nodes in the neighbourhood.

Loop's contribution was to determine what new values should be given to the mesh nodes in order to ensure convergence on a C^1 continuous surface for arbitrary topology. For a regular mesh where all nodes have a valance of six† the subdivision coefficients are illustrated for vertex insertion in Figure 5.2.

On recursive application of these weightings a surface converges on a smooth geometry. This is shown in Figure 5.3. The torus is specifically shown here as its topology is amenable to regular meshing, having a Euler characteristic $\chi = 0$. Inspecting the nodes, each one has a regular valance of six.

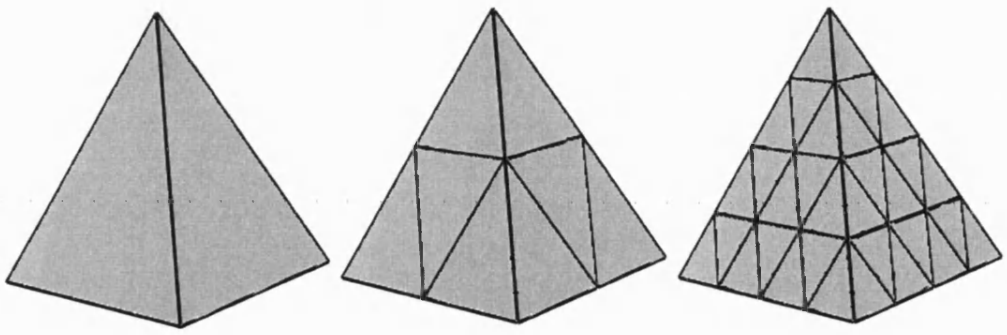


Figure 5.1 Linear subdivision of a tetrahedral surface

† The valance is a measure of the number of mesh edges meeting at a point. A regular triangular mesh is one in which every node has exactly six mesh edges converging on it. An extraordinary node has a valance other than six.

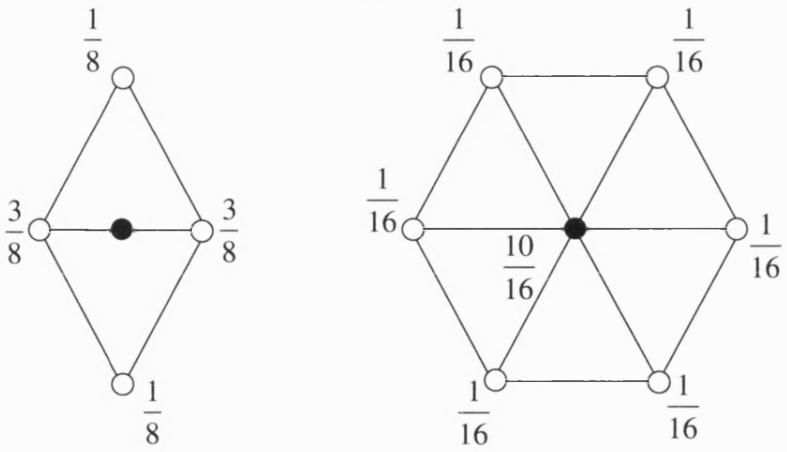


Figure 5.2 Subdivision coefficients for a three-directional box spline, reproduced from Zorin and Schröder[39]

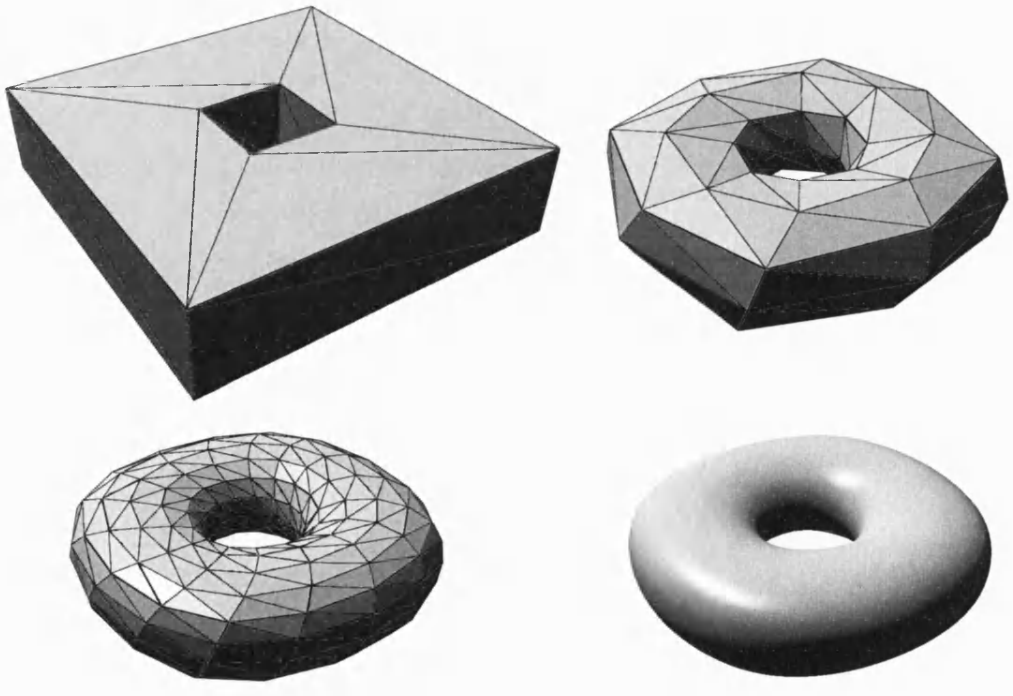


Figure 5.3 Loop subdivision applied to a toroidal topology mesh

Loop's next step was to generalise these coefficients to accommodate arbitrary valances ($k \neq 6$), as shown in Figure 5.4. Loop originally suggested the value

$$\beta = \frac{1}{k}\left[\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{k}\right)^2\right]. \qquad [5.1]$$

However more recently Warren[129] suggested a simpler value for β which gives comparably smooth results and avoids the use of trigonometric functions in the computation,

$$\beta = \frac{3}{8k} \text{ for } k > 3 \quad \beta = \frac{3}{16} \text{ for } k = 3\ . \qquad [5.2]$$

Warren's weightings are applied to the cube in Figure 5.5, whose topology requires these extraordinary valances.

Elements on the boundary can be handled a number of ways depending on the boundary condition required. One-dimensional subdivision can be applied along the curve length of the boundary or, if for instance tangency at the boundary is required, *ghost* or *imaginary* nodes can be added outside of the domain which can then control the conditions on the boundary[84]. This will be discussed in detail in part three when enforcing the boundary conditions of the differential equations.

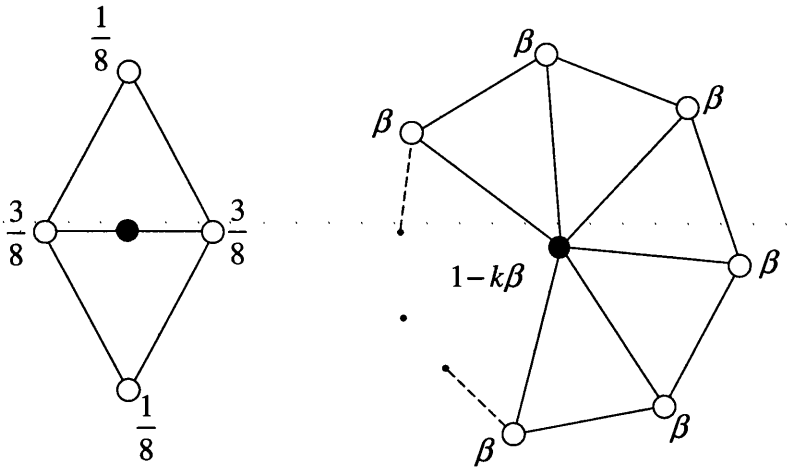


Figure 5.4 Subdivision masks for the Loop scheme, reproduced from Zorin and Schröder[39]

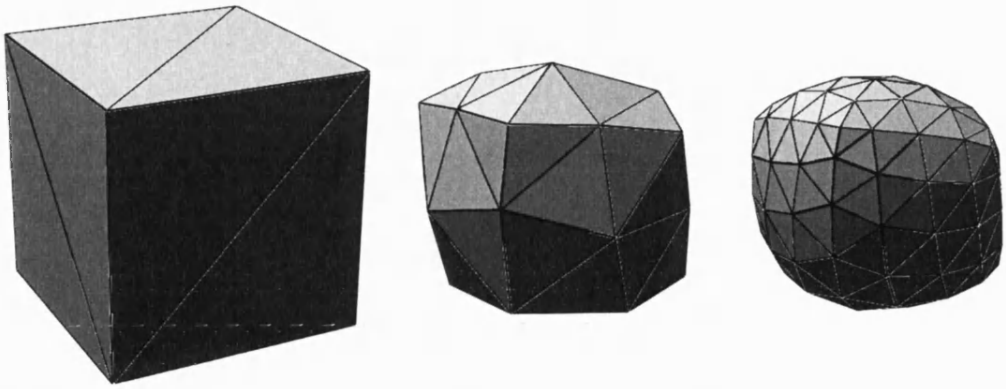


Figure 5.5 Loop subdivision applied to a spherical topology mesh

Like B-splines the Loop subdivision method cannot reproduce the conic sections. Loop is an example of a stationary subdivision scheme in which the subdivision rules applied to the vertices are a constant for each level of subdivision. Methods to reproduce the conic sections have been developed, using non-stationary subdivision schemes, where the rules are dependent on the level of recursion[58, 59]. Using Loop, circular boundary geometries cannot be exactly reproduced. However for our implementation a sufficiently fine mesh of data points shall be applied at the boundary to catch the perturbations in the stream function and therefore the difference between our approximation and a true circular arc shall be negligible.

5.2 Exact interrogation of subdivision surfaces

Although one can go through this process of recursively creating finer and finer meshes, possibly for visualisation purposes, in order to utilise subdivision as an approach for discretising the governing differential equations the limit surface needs to be explicitly interrogable.

The mask in Figure 5.4 shows how the position of each point is determined by its immediate surrounding neighbours, conventionally called the point's one-ring. A triangular patch of surface is therefore defined by each of its vertices and their

one-rings. Figure 5.6 shows a regular patch of surface and its twelve control points governing its geometry.

5.2.1 Subdivision shape functions

In *Evaluation of Loop subdivision surfaces*, Stam[86] shows the position vector $\mathbf{x}$ of a point on the limit surface for a regular patch can be defined using twelve shape functions. The shape functions are written in full in Appendix B, along with their first and second derivatives, which will be of use later. However the twelve expressions can be reduced to the four independent functions below, taking for each, the values $i = 0, 1, 2$.

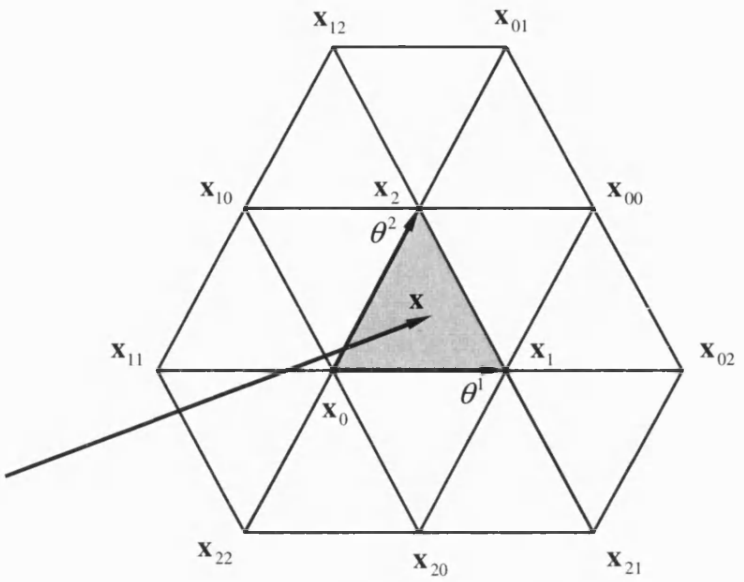
$$\begin{aligned}
N^{i} &= \frac{1}{12}\left(6u_0^{\,4} + 24u_0^{\,3}u_2 + 24u_0^{\,2}u_2^{\,2} + 8u_0u_2^{\,3} + u_2^{\,4} + 24u_0^{\,3}u_1\right. \\
&\quad + 60u_0^{\,2}u_1u_2 + 36u_0u_1u_2^{\,2} + 6u_1u_2^{\,3} + 24u_0^{\,2}u_1^{\,2} + 36u_0u_1^{\,2}u_2 \\
&\quad \left. + 12u_1^{\,2}u_2^{\,2} + 8u_0u_1^{\,3} + 6u_1^{\,3}u_2 + u_1^{\,4}\right), \\
N^{i0} &= \frac{1}{12}\left(2u_0u_2^{\,3} + u_2^{\,4} + 6u_0u_1u_2^{\,2} + 6u_1u_2^{\,3} + 6u_0u_1^{\,2}u_2\right. \\
&\quad \left. + 12u_1^{\,2}u_2^{\,2} + 2u_0u_1^{\,3} + 6u_1^{\,3}u_2 + u_1^{\,4}\right), \\
N^{i1} &= \frac{1}{12}\left(2u_1^{\,3}u_2 + u_1^{\,4}\right), \\
N^{i2} &= \frac{1}{12}\left(u_2^{\,4} + 2u_1u_2^{\,3}\right),
\end{aligned} \qquad [5.3]$$

where using barycentric coordinates (u, v, w) the following conditions apply

$$\begin{aligned}
&\text{when } i = 0,\ u_0 = u,\ u_1 = v \text{ and } u_2 = w, \\
&\text{when } i = 1,\ u_0 = v,\ u_1 = w \text{ and } u_2 = u, \\
&\text{when } i = 2,\ u_0 = w,\ u_1 = u \text{ and } u_2 = v,
\end{aligned}$$

and the curvilinear coordinates θ^1 and θ^2 conform to the following expressions

$$\begin{aligned}
v &= \theta^1, \\
w &= \theta^2, \\
u &= 1 - \theta^1 - \theta^2.
\end{aligned}$$



5.6 Twelve control points governing a regular subdivision patch

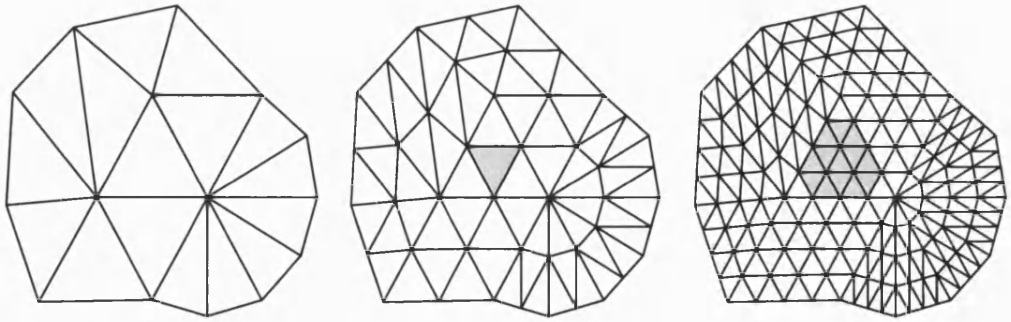
The numbering convention adopted here, shown in Figure 5.6, has been devised to exploit the rotational symmetry of the problem, otherwise the twelve basis functions shown are as given by Stam[86]. Note all superscripts on the right-hand side of the equations represent exponents.

The position vector $\mathbf{x}$ of an arbitrary point on the limit surface is then given using these shape functions as

$$\mathbf{x}(\theta^1,\theta^2) = N^I(\theta^1,\theta^2)\mathbf{x}_I . \qquad [5.4]$$

where I takes the twelve possible values given above.

It shall be shown in chapter six that equation [5.4] is a useful form of continuum discretisation compatible with the finite element method. However it is clear that these twelve shape functions may only be applied to regular patches.



5.8 The area of limit surface determinable (highlighted in grey) in an irregular element using shape functions, for original mesh and subsequent levels of subdivision

5.2.2 Generation of regular surface patches

Loop's rules show topological subdivision generates areas of regular mesh. Inspection of Figure 5.5 shows the extraordinary valance at the apex of the original cube is increasingly isolated by expanses of nodes of valance six as the subdivision progresses. If the triangle we are interested in is not regular then a process of topological subdivision, using the masks in Figure 5.4, can be applied until the point of interest falls within the domain of a regular patch. The process is shown in Figure 5.8. This newly generated triangular patch can then be interrogated and properties of the limit surface obtained using equation [5.4].

5.3 Conclusions

The objective of this chapter was to derive an approach for defining C^1 continuous surfaces on an arbitrary triangular mesh. The existing approach proposed by Loop[38] has been identified.

Evaluating a surface in this way gives C^2-continuity in areas of regularity and C^1-continuity at the extraordinary valance nodes[39, 88].

This procedure will now be extended in the next part of this thesis as it is put forward as a candidate for discretising the stream function Navier-Stokes equation.

Part three

Derivation and implementation of theory

Chapter six

Formulation of the numerical method

6.0 Introduction

In this chapter the finite element approach is applied to the stream function equation previously derived in chapter four. The necessity for a C^1 continuous shape function is then demonstrated. To efficiently model this governing equation a highly irregular underlying mesh will be required. The suitability of the existing Loop subdivision method as an approach to surface description and hence as a finite element is therefore shown.

An irregular topology subdivision finite element is then developed through extension of the methodologies set out in chapter five. A novel generalised subdivision matrix for an arbitrary configuration of control points is derived. This is applied to an arbitrary patch of surface. This new subdivision element is then exploited with the descretised governing stream function equation.

Dynamic relaxation as an iterative solution approach is introduced to be implemented in the algorithms of chapters seven and eight enabling efficient solution of the sparse matrices.

The objective of this chapter is therefore to derive the necessary methodologies to solve the governing equations in the following chapters.

6.1 Weighted residual formulation of the stream function

The fourth-order stream function equation in general curvilinear tensor notation, as given in the chapter four, is reproduced as

$$\left(\frac{\partial}{\partial t}+\varepsilon^{\gamma\eta}\nabla_{\eta}\psi\nabla_{\gamma}-\frac{\mu}{\rho}a^{\gamma\eta}\nabla_{\gamma\eta}\right)\left(a^{\alpha\beta}\nabla_{\alpha\beta}\psi\right)=0\,. \qquad [4.37]$$

Using a finite element methodology

$$\psi\left(\theta^1,\theta^2,t\right)\approx N^I\left(\theta^1,\theta^2\right)\psi_I(t), \qquad [6.1]$$

we can express equation [4.37] in a discretised form.

A point within a typical element is evaluated as a summation of contributions from the control points in its neighbourhood (see Figure 6.1). The shape functions N^I are defined spatially with respect to coordinates θ^1 and θ^2, such that $N^I=0$ for nodes which do not influence the element.

We are now assuming an approximate solution of ψ, requiring that in order to satisfy any boundary conditions, equation [4.37] can no longer be exactly equal to zero.

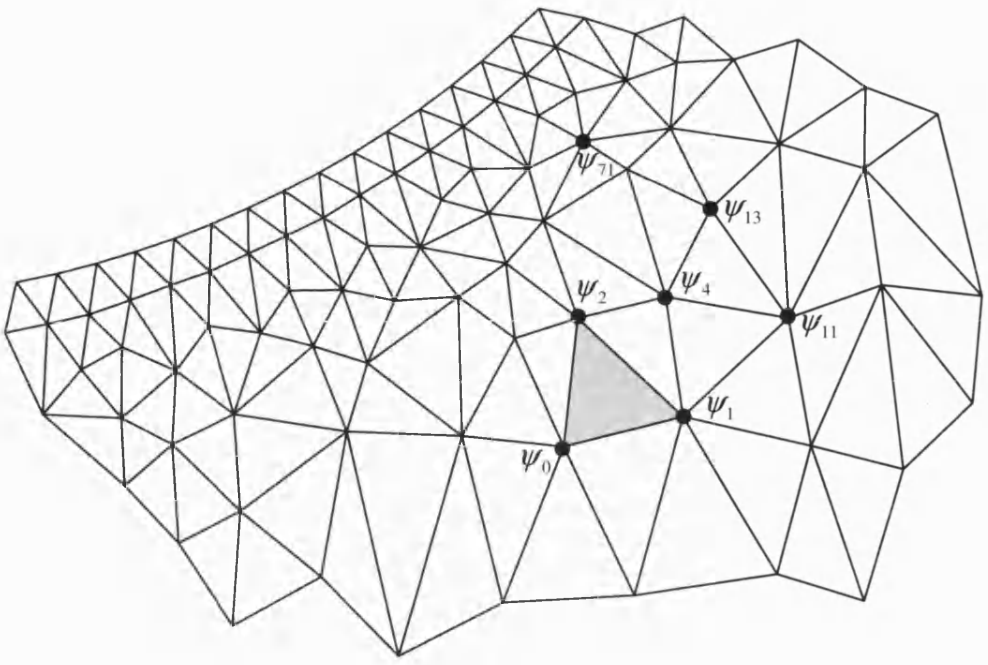


Figure 6.1 Portion of an unstructured triangular finite element mesh, highlighting one element and an arbitrary collection of mesh vertices

Thus we have

$$\left(\dot{\psi}_J + \varepsilon^{\gamma\eta}\psi_J\psi_K\nabla_\eta N_K\nabla_\gamma - \frac{\mu}{\rho}\psi_J a^{\gamma\eta}\nabla_{\gamma\eta}\right)\left(a^{\alpha\beta}\nabla_{\alpha\beta}N^J\right) \neq 0, \qquad [6.2]$$

in which $\dot{\psi}_J = \dfrac{d\psi_J}{dt}$. Note that ψ_J is a function of time only.

Equation [6.2] now represents an error or residual, which can be written in a weighted residual formulation using weighting functions $W^I\left(\theta^1, \theta^2\right)$.

This gives

$$\int_\Omega W^I\left(\dot{\psi}_J + \varepsilon^{\gamma\eta}\psi_J\psi_K\nabla_\eta N_K\nabla_\gamma - \frac{\mu}{\rho}\psi_J a^{\gamma\eta}\nabla_{\gamma\eta}\right)\left(a^{\alpha\beta}\nabla_{\alpha\beta}N^J\right)d\Omega = 0,$$

for all I, where the integration takes place over the entire region Ω.

Taking the time dependent terms out of the integration, we obtain

$$\dot{\psi}_J \int W^I a^{\alpha\beta} \nabla_{\alpha\beta} N^J d\Omega + \psi_J \psi_K \int W^I \varepsilon^{\gamma\eta} a^{\alpha\beta} \nabla_\eta N^J \nabla_{\gamma\alpha\beta} N^K d\Omega$$
$$-\frac{\mu}{\rho} \psi_J \int W^I a^{\gamma\eta} a^{\alpha\beta} \nabla_{\gamma\eta\alpha\beta} N^J d\Omega = 0. \qquad [6.3]$$

Denoting the three integrals constituting equation [6.3] as I_1, I_2 and I_3 respectively, these equations can be simplified.

An integration by parts can be performed on the first integral giving,

$$I_1 = \int W^I a^{\alpha\beta} \nabla_{\alpha\beta} N^J d\Omega = \int a^{\alpha\beta} \nabla_\alpha \left(W^I \nabla_\beta N^J \right) d\Omega - \int a^{\alpha\beta} \nabla_\alpha W^I \nabla_\beta N^J d\Omega .$$

Using the divergence theorem[120, 130]

$$\int \nabla_i q^i d\Omega = \oint q^i n_i dl .$$

in which n_i are the components of the outwards pointing unit normal and dl is an element of boundary, the integral becomes

$$I_1 = \oint a^{\alpha\beta} W^I \nabla_\beta N^J n_\alpha dl - \int a^{\alpha\beta} \nabla_\alpha W^I \nabla_\beta N^J d\Omega .$$

Now we can reasonably impose the restriction on our choice of weighting function that $W^I = 0$ on the boundary of integration. The integration therefore simplifies to

$$I_1 = \int W^I a^{\alpha\beta} \nabla_{\alpha\beta} N^J d\Omega = -\int a^{\alpha\beta} \nabla_\alpha W^I \nabla_\beta N^J d\Omega . \qquad [6.4]$$

The result of this process is that we have now lowered the order of the derivatives of our shape functions, N^J, relaxing our continuity requirements.

A similar process of simplification can be performed on each of the other two integrals, I_2 and I_3,

$$I_2 = \int W^I \varepsilon^{\gamma\eta} a^{\alpha\beta} \nabla_\eta N^J \nabla_{\gamma\alpha\beta} N^K d\Omega,$$

Now, using $\nabla_{\gamma\eta} = \nabla_{\eta\gamma}$, we can express I_2 as

$$\begin{aligned} I_2 &= \int W^I \varepsilon^{\gamma\eta} a^{\alpha\beta} \nabla_\gamma \left(\nabla_\eta N^J \nabla_{\alpha\beta} N^K\right) d\Omega, \\ &= \int \nabla_\gamma \left(W^I \varepsilon^{\gamma\eta} a^{\alpha\beta} \nabla_\eta N^J \nabla_{\alpha\beta} N^K\right) d\Omega - \int \varepsilon^{\gamma\eta} a^{\alpha\beta} \nabla_\gamma W^I \nabla_\eta N^J \nabla_{\alpha\beta} N^K d\Omega, \\ &= \oint W^I \varepsilon^{\gamma\eta} a^{\alpha\beta} \nabla_\eta N^J \nabla_{\alpha\beta} N^K n_\gamma dl - \int \varepsilon^{\gamma\eta} a^{\alpha\beta} \nabla_\gamma W^I \nabla_\eta N^J \nabla_{\alpha\beta} N^K d\Omega, \end{aligned}$$

Thus we can say

$$I_2 = -\int \varepsilon^{\gamma\eta} a^{\alpha\beta} \nabla_\gamma W^I \nabla_\eta N^J \nabla_{\alpha\beta} N^K d\Omega \,. \qquad [6.5]$$

Finally

$$\begin{aligned} I_3 &= \int W^I a^{\gamma\eta} a^{\alpha\beta} \nabla_{\gamma\eta\alpha\beta} N^J d\Omega, \\ &= \int a^{\gamma\eta} a^{\alpha\beta} \nabla_\gamma \left(W^J \nabla_{\eta\alpha\beta} N^J\right) d\Omega - \int \nabla_\gamma W^I \nabla_{\eta\alpha\beta} N^J d\Omega, \\ &= \int a^{\gamma\eta} a^{\alpha\beta} \nabla_\gamma \left(W^J \nabla_{\eta\alpha\beta} N^J\right) d\Omega - \int a^{\gamma\eta} a^{\alpha\beta} \nabla_{\gamma\eta} \left(W^I \nabla_{\alpha\beta} N^J\right) d\Omega + \\ &\qquad \int a^{\gamma\eta} a^{\alpha\beta} \nabla_{\gamma\eta} W^I \nabla_{\alpha\beta} N^J d\Omega, \\ &= \oint a^{\gamma\eta} a^{\alpha\beta} W_J \nabla_{\eta\alpha\beta} N^J n_\gamma dl - \oint a^{\gamma\eta} a^{\alpha\beta} \nabla_\gamma \left(W^I \nabla_{\alpha\beta} N^J\right) n_\gamma dl + \\ &\qquad \int a^{\gamma\eta} a^{\alpha\beta} \nabla_{\gamma\eta} W^I \nabla_{\alpha\beta} N^J d\Omega. \end{aligned}$$

In addition to the condition of $W^I = 0$ on the boundary, if we employ a C^1 continuous shape function we also have the relationship $\nabla_\gamma W^I = 0$ defined on the boundary of integration. The third integral therefore simplifies to

$$I_3 = \int a^{\gamma\eta} \nabla_{\gamma\eta} W^I a^{\alpha\beta} \nabla_{\alpha\beta} N^J d\Omega. \qquad [6.6]$$

Thus, substituting equations [6.4], [6.5] and [6.6] back into equation [6.3] we obtain what is known as the *weak form* of the differential equations,

$$\begin{aligned}&\dot{\psi}_J \int a^{\alpha\beta}\nabla_\alpha W^I \nabla_\beta N^J d\Omega + \\ &\psi_J\psi_K \int \varepsilon^{\gamma\eta}\nabla_\gamma W^I \nabla_\eta N^J a^{\alpha\beta}\nabla_{\alpha\beta}N^K d\Omega + \\ &\frac{\mu}{\rho}\psi_J \int a^{\gamma\eta}\nabla_{\gamma\eta} W^I a^{\alpha\beta}\nabla_{\alpha\beta}N^J d\Omega = 0.\end{aligned} \qquad [6.7]$$

Expressing the problem in the weak form in this way has distinct advantages. The original partial differential equation was seen to contain fourth-order derivatives. Now looking at equation [6.7] this has been reduced to a second-order equation. This means the continuity requirements of the shape functions has been relaxed so only second derivatives are required; only continuity of slope between elements, so called C^1 continuity, is needed.

The same level of continuity is also now required by the weighting functions. Any function can be chosen for W^I, however the implementation can be simplified through the widely used *Galerkin* approach setting $W^I = N^I$. This weighted residual Galerkin formulation is equivalent to a virtual work formulation[83, 131].

Equation [6.7] now becomes

$$\begin{aligned}&\dot{\psi}_J \int a^{\alpha\beta}\nabla_\alpha N^I \nabla_\beta N^J \sqrt{a}d\theta^1 d\theta^2 + \\ &\psi_J\psi_K \int \varepsilon^{\gamma\eta}\nabla_\gamma N^I \nabla_\eta N^J a^{\alpha\beta}\nabla_{\alpha\beta}N^K \sqrt{a}d\theta^1 d\theta^2 + \\ &\frac{\mu}{\rho}\psi_J \int a^{\gamma\eta}\nabla_{\gamma\eta} N^I a^{\alpha\beta}\nabla_{\alpha\beta}N^J \sqrt{a}d\theta^1 d\theta^2 = 0.\end{aligned} \qquad [6.8]$$

This is the discretised stream function equation, the solution of which forms the focus of this thesis.

6.2 Requirements of the stream surface shape function

Interestingly, as Zienkiewicz[83, 131] notes is often the case, the weak formulation of the differential equation is physically more intuitive than the original. Looking at the third term of equation [6.8], this is the weak form of the fourth derivative of ψ, $\nabla^4\psi$, analogous to plate bending. This *excessive smoothness* has now been relaxed through the integration by parts and this term can now be interpreted as the square mean curvature.

$$a^{\alpha\beta}\nabla_{\alpha\beta}N^I = 2H\,.$$

This shows the requirement of the shape functions N^I to be C^1 continuous.

C^1 continuity across patches of a function ψ requires, by definition, ψ and $\partial\psi/\partial\theta^1$ and $\partial\psi/\partial\theta^2$ to be uniquely defined along the interface between abutting elements. As Zienkiewicz and Taylor[123] describe, matching these three expressions along edges results in inconsistencies at the common corner nodes. The identity

$$\frac{\partial^2\psi}{\partial\theta^1\partial\theta^2} = \frac{\partial^2\psi}{\partial\theta^2\partial\theta^1}$$

is not necessarily met.† As a consequence, for an arbitrarily orientated element, *all* second derivatives at the node need to be specified. Therefore implementing simple polynomial shape functions must include these higher order derivatives or alternatively be non-conforming, with slopes normal to the element edges discontinuous[133].

† Similarly in differential geometry we have the requirement of the Gauss and Codazzi equations. As shown in chapter two we have six variables $a_{\alpha\beta}$ and $b_{\alpha\beta}$ derived from the three spatial coordinates of the position vector $x^k\left(\theta^1,\theta^2\right)$. Thus the equation of Gauss and the two Codazzi equations ensure the surface *fits together*[132]

A C^0 continuous discretisation, in other words no continuity of slope, gives a faceted surface of flat triangles. This is sufficient for modelling pure tensile surfaces, soap film and minimal surfaces, however for our stream function surface these hinges or kinks in the manifold would equate to an infinite shear strain at the interface between elements and a constant slope and therefore velocity within each element. This would not constitute a satisfactory solution. Thus the proposal here is to use subdivision shape functions which through the sole consideration of displacements achieve C^1 continuity between curvilinear elements.

6.3 Discretisation of the stream function equation with subdivision shape functions

In this thesis the triangular subdivision methodology proposed by Loop[38] is adopted. Loop's recursive topological refinement process converging to a smooth limit surface was described in chapter five, as well as Stam's[86] procedure of evaluating the limit surface through the use of shape functions. These methodologies shall be extended for application in this thesis.

Obviously one can form a mesh out of any polygonal shape, however the most commonly used meshes are those derived from the triangle or rectangle. A triangular discretisation has the obvious advantage over other tessellations in that irregular boundary geometries can be accommodated much more easily.

An example of a triangular mesh conforming to an arbitrary form is given in Figure 6.2. Here the intricate boundary geometry is easily meshed using an irregular triangulation. Small patches of regularity can be seen in the mesh, but in general, any given triangle has vertices of extraordinary valance.

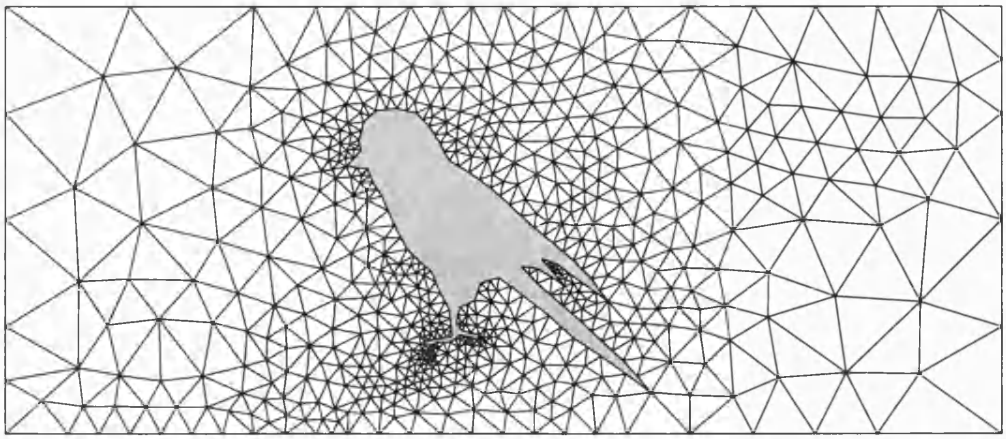


Figure 6.2 A constrained Delaunay triangulation driven by boundary geometry

The finite element meshes used in this thesis are all generated using Triangle[134-136] a freely available Delaunay triangulator. Here the input for the mesh is the boundary geometry. The output from Triangle comes in the form of a list of nodes, triangular elements and lines. A program was then devised in order to translate this data into element-node relations required for the subdivision finite element. Search algorithms were used to assign each mesh node with its surrounding one-ring of nodes in anti-clockwise orientation. From this information the degrees of freedom for each element could then be associated.

For a regular element the numbering convention given in chapter five, Figure 5.6 is used. This gives the following list of twelve position vectors, transposed from the column vector for convenience.

$$\begin{bmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_{00} & \mathbf{x}_{01} & \mathbf{x}_{02} & \mathbf{x}_{10} & \mathbf{x}_{11} & \mathbf{x}_{12} & \mathbf{x}_{20} & \mathbf{x}_{21} & \mathbf{x}_{22} \end{bmatrix}$$

For elements with an extraordinary vertex, the following node notation has been devised, illustrated in Figure 6.3. Here for clarity only one irregular node is shown, however the convention stands for an element with any configuration of nodes.

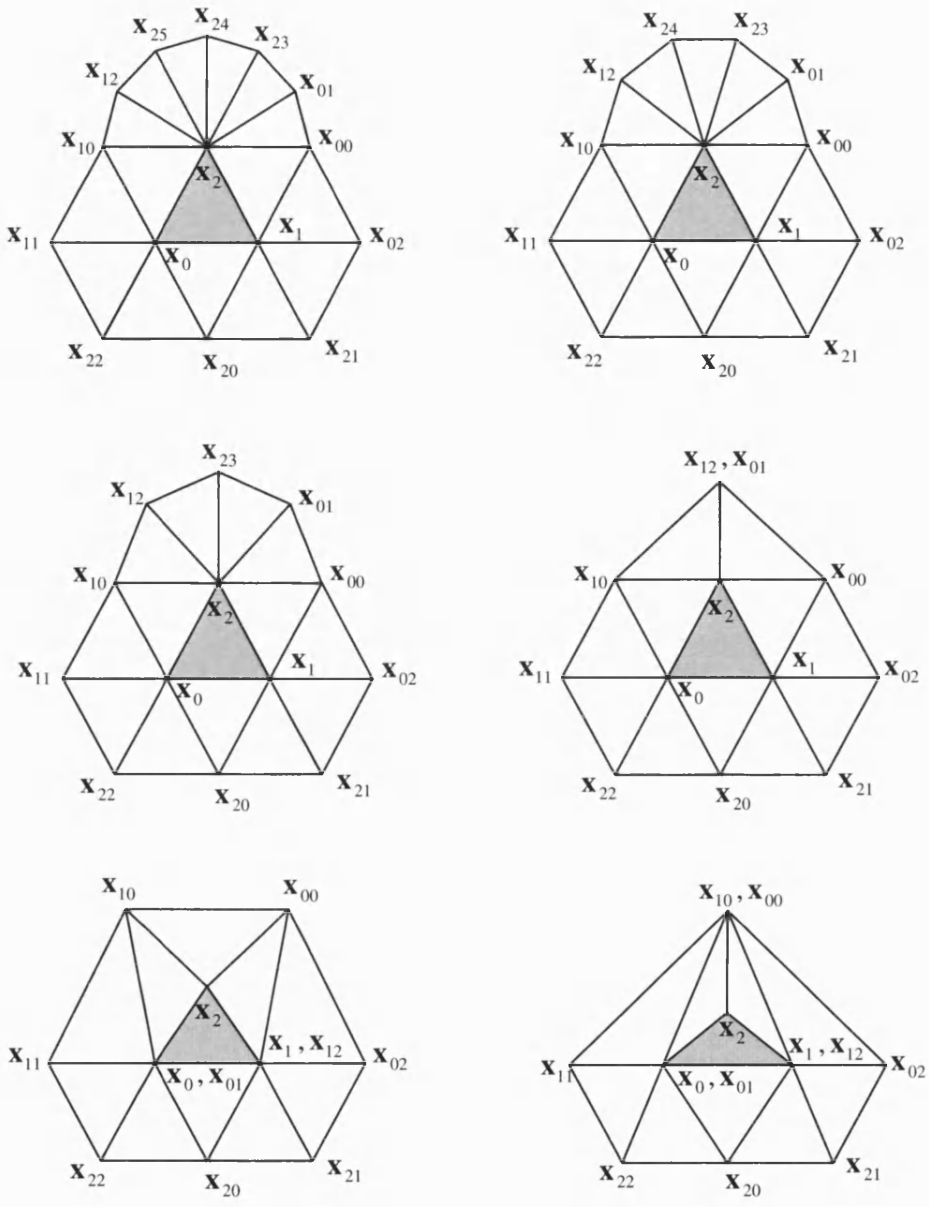


Figure 6.3 Numbering convention for control points related to irregular elements

Expansion and contraction of the valance of vertex $\mathbf{x}_2$, illustrating the numbering system adopted for extraordinary elements is shown in Figure 6.3. Therefore the column of position vectors in general is not limited to just twelve entries.

The Loop subdivision shape functions given in Appendix B can only be applied to regular elements with exactly twelve control points. As shown in chapter five any

element can be subdivided to create a patch of regular mesh describable with the Loop shape functions. For this work a simple one-point quadrature numerical integration as implemented by Cirak *et al.*[84, 89, 111] is used. This has been shown through numerical tests to be sufficient for convergence. However the process described could be expanded to accommodate more complex quadrature methods. For the general case of an arbitrary mesh the given element must be subdivided once. The newly generated central quarter of the element is now completely determined allowing evaluation of a point at the centre of the element. We therefore have a two level hierarchy of controls points, as shown in Figure 6.4, which we shall call the subdivided local and the global points, the former being dependant on the latter. The relationship between these two sets of points is given by the Loop subdivision masks as given in chapter five. These general rules for each node can be formulated into a rectangular matrix relating the control points for the element in question.

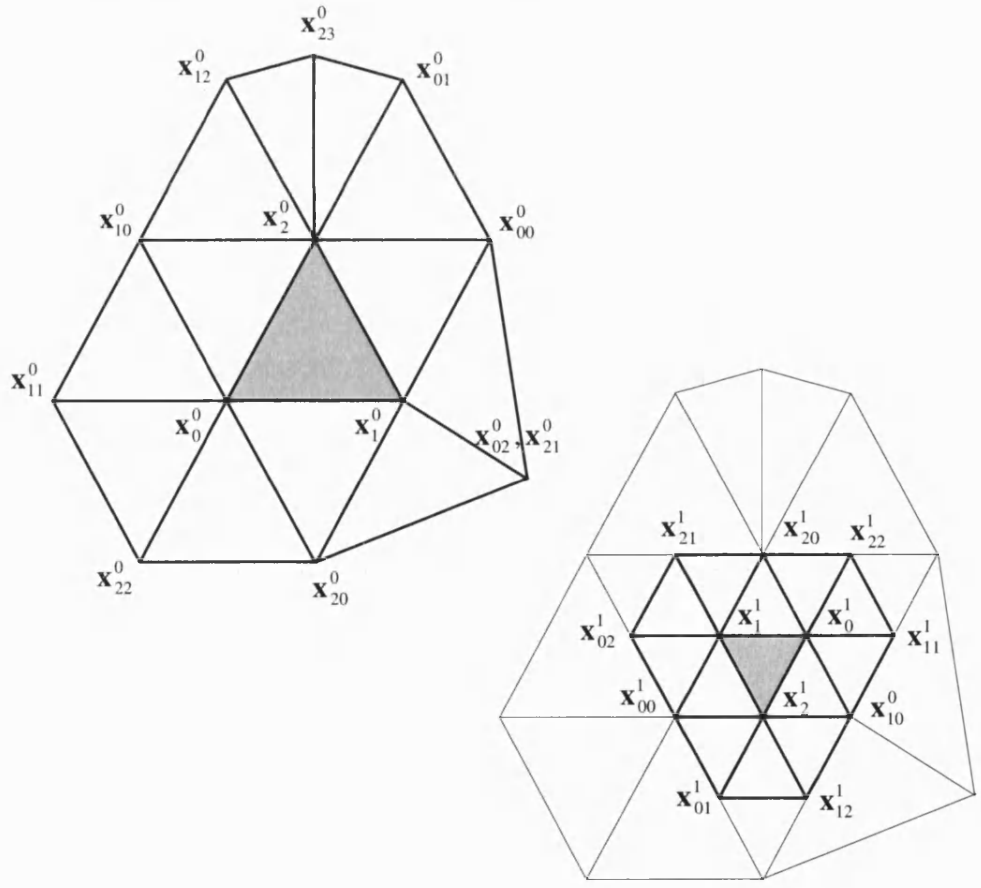


Figure 6.4 An irregular triangular finite element shown in plan, illustrating the global and local control points

Starting with the case of a regular element (obviously this in itself would not require further subdivision in order to apply the twelve shape functions. However the arbitrary case can be understood more easily as a small step from this simplified arrangement), the regular subdivision matrix is given by

$$\begin{bmatrix} \mathbf{x}_0^1 \\ \mathbf{x}_1^1 \\ \mathbf{x}_2^1 \\ \mathbf{x}_{00}^1 \\ \mathbf{x}_{10}^1 \\ \mathbf{x}_{20}^1 \\ \mathbf{x}_{01}^1 \\ \mathbf{x}_{11}^1 \\ \mathbf{x}_{21}^1 \\ \mathbf{x}_{02}^1 \\ \mathbf{x}_{12}^1 \\ \mathbf{x}_{22}^1 \end{bmatrix} = \begin{bmatrix} \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{3}{8} & \frac{1}{8} & \frac{3}{8} & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{10}{16} & \frac{1}{16} & \frac{1}{16} & 0 & \frac{1}{16} & \frac{1}{16} & 0 & \frac{1}{16} & 0 & 0 & 0 & \frac{1}{16} \\ \frac{1}{16} & \frac{10}{16} & \frac{1}{16} & \frac{1}{16} & 0 & \frac{1}{16} & 0 & 0 & \frac{1}{16} & \frac{1}{16} & 0 & 0 \\ \frac{1}{16} & \frac{1}{16} & \frac{10}{16} & \frac{1}{16} & \frac{1}{16} & 0 & \frac{1}{16} & 0 & 0 & 0 & \frac{1}{16} & 0 \\ \frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & \frac{3}{8} & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} \\ 0 & \frac{3}{8} & \frac{1}{8} & \frac{3}{8} & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} & 0 & 0 \\ \frac{1}{8} & 0 & \frac{3}{8} & 0 & \frac{3}{8} & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} & 0 \\ \frac{3}{8} & 0 & \frac{1}{8} & 0 & \frac{3}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 \\ \frac{1}{8} & \frac{3}{8} & 0 & 0 & 0 & \frac{3}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 \\ 0 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0^0 \\ \mathbf{x}_1^0 \\ \mathbf{x}_2^0 \\ \mathbf{x}_{00}^0 \\ \mathbf{x}_{10}^0 \\ \mathbf{x}_{20}^0 \\ \mathbf{x}_{01}^0 \\ \mathbf{x}_{11}^0 \\ \mathbf{x}_{21}^0 \\ \mathbf{x}_{02}^0 \\ \mathbf{x}_{12}^0 \\ \mathbf{x}_{22}^0 \end{bmatrix}$$

A generalised subdivision matrix can be written for any arbitrary configuration of mesh using the notation convention for extraordinary nodes as in Figure 6.3,

$$\begin{bmatrix} \mathbf{x}_0^1 \\ \mathbf{x}_1^1 \\ \mathbf{x}_2^1 \\ \mathbf{x}_{00}^1 \\ \mathbf{x}_{10}^1 \\ \mathbf{x}_{20}^1 \\ \mathbf{x}_{01}^1 \\ \mathbf{x}_{11}^1 \\ \mathbf{x}_{21}^1 \\ \mathbf{x}_{02}^1 \\ \mathbf{x}_{12}^1 \\ \mathbf{x}_{22}^1 \end{bmatrix} = \begin{bmatrix} \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \frac{3}{8} & \frac{1}{8} & \frac{3}{8} & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \frac{10}{16} & \frac{3}{8k_0} & \frac{3}{8k_0} & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{3}{8k_1} & \frac{10}{16} & \frac{3}{8k_1} & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{3}{8k_2} & \frac{3}{8k_2} & \frac{10}{16} & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & \frac{3}{8} & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} & \dots & 0 \\ 0 & \frac{3}{8} & \frac{1}{8} & \frac{3}{8} & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} & 0 & 0 & \dots & 0 \\ \frac{1}{8} & 0 & \frac{3}{8} & 0 & \frac{3}{8} & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} & 0 & \dots & 0 \\ \frac{3}{8} & 0 & \frac{1}{8} & 0 & \frac{3}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & \dots & 0 \\ \frac{1}{8} & \frac{3}{8} & 0 & 0 & 0 & \frac{3}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0^0 \\ \mathbf{x}_1^0 \\ \mathbf{x}_2^0 \\ \mathbf{x}_{00}^0 \\ \mathbf{x}_{10}^0 \\ \mathbf{x}_{20}^0 \\ \mathbf{x}_{01}^0 \\ \mathbf{x}_{11}^0 \\ \mathbf{x}_{21}^0 \\ \mathbf{x}_{02}^0 \\ \mathbf{x}_{12}^0 \\ \mathbf{x}_{22}^0 \\ \vdots \\ \vdots \end{bmatrix} \quad [6.9]$$

here k_0, k_1 and k_2 equal the valances of vertices $\mathbf{x}_0^0$, $\mathbf{x}_1^0$ and $\mathbf{x}_2^0$ respectively.

In order to expose the symmetry between the subdivision matrices of any arbitrary topology element (and therefore simplifying the implementation), coincident vertices, occurring due to contraction of a valance, are included in the column vectors of control points in each reference. It is worth noting that these column vectors are just references or *pointers* to the actual nodes. Where nodes are coincident the references will point to the same coordinates. The actual number of dependant nodes thus must take into account these equalities. Expansion of a valance is handled in a straight forward manner with its impact only being seen in row four, five or six of the subdivision matrix corresponding to the new node at that original apex.

Dropping the superscripts 0 or 1 on the control points $\mathbf{x}^0$, $\mathbf{x}^1$ to denote the level of subdivision we shall denote the global control points for an element as $\mathbf{x}_P$, using Latin capital letters. The local control points $\mathbf{x}_\Pi$ shall only use Greek capital letters. Therefore the Greek index shall always take twelve values whereas the number of values taken by the Latin index shall vary depending on the irregularity of the element of interest. For a regular element the number of global control points shall also be twelve but for an irregular element it may be more. Whether the index is Latin or Greek now indicates whether we are concerned with the global or local nodes.

Using this notation the relation between the two subdivision levels of nodes can therefore be written as

$$\mathbf{x}_\Pi = S_\Pi^P \mathbf{x}_P . \qquad [6.10]$$

The subdivision matrix S_Π^P can be therefore be considered as a form of transformation matrix, transforming from global to local degrees of freedom

$$S_\Pi^P = \frac{\partial \mathbf{x}_\Pi}{\partial \mathbf{x}_P} .$$

This notation is similar to the method adopted by Dirac[137] for writing transformation of coordinates, where the transformation is attached to the suffix as opposed to the main symbol, as shown by

$$a^{\alpha'\beta'} = \theta^{\alpha'}_{,\lambda}\theta^{\beta'}_{,\mu}a^{\lambda\mu}.$$

With the degrees of freedom now associated with each element we can return to the discretised stream function equation. The terms on the right hand side of this equation can only be evaluated within a regular patch of the limit surface. However we can now use the general subdivision matrix to relate an arbitrary point to the primary unknowns, the global degrees of freedom, ψ_P

$$\begin{aligned}\psi(\theta^1,\theta^2,t) &= N^{\Pi}(\theta^1,\theta^2)\psi_{\Pi}(t),\\ &= N^{\Pi}(\theta^1,\theta^2)S^P_{\Pi}\psi_P(t).\end{aligned} \qquad [6.11]$$

Thus the notation employed in the weighted residual formulation can be clarified for this specific case of a subdivision element on an arbitrary mesh.

The previously derived stream function equation [6.8] can then be rewritten as

$$\begin{aligned}&\dot{\psi}_J\int a^{\alpha\beta}\nabla_\alpha N^{\Pi}S^I_{\Pi}\nabla_\beta N^{\Theta}S^J_{\Theta}\sqrt{a}d\theta^1d\theta^2 +\\ &\psi_J\psi_K\int\varepsilon^{\gamma\eta}\nabla_\gamma N^{\Pi}S^I_{\Pi}\nabla_\eta N^{\Sigma}S^J_{\Sigma}a^{\alpha\beta}\nabla_{\alpha\beta}N^{\Theta}S^K_{\Theta}\sqrt{a}d\theta^1d\theta^2 +\\ &\frac{\mu}{\rho}\psi_J\int a^{\gamma\eta}\nabla_{\gamma\eta}N^{\Pi}S^I_{\Pi}a^{\alpha\beta}\nabla_{\alpha\beta}N^{\Theta}S^J_{\Theta}\sqrt{a}d\theta^1d\theta^2 = 0.\end{aligned} \qquad [6.12]$$

Here the shape functions all have Greek indices, indicating they are evaluated locally. However through the application of the subdivision matrices the equation still references the global degrees of freedom.

Equation [6.12] can be simplified by writing

$$\dot{\psi}_J D^{IJ} + \psi_J \psi_K C^{IJK} + \frac{\mu}{\rho} \psi_J K^{IJ} = 0, \qquad [6.13]$$

where

$$\begin{aligned} D^{PQ} &= D^{\Pi\Theta} S^P_\Pi S^Q_\Theta, \\ C^{PQS} &= C^{\Pi\Theta\Sigma} S^P_\Pi S^Q_\Theta S^S_\Sigma, \\ K^{PQ} &= K^{\Pi\Theta} S^P_\Pi S^Q_\Theta, \end{aligned} \qquad [6.14]$$

and

$$\begin{aligned} D^{\Pi\Theta} &= \int a^{\alpha\beta} \nabla_\alpha N^\Pi \nabla_\beta N^\Theta \sqrt{a} d\theta^1 d\theta^2, \\ C^{\Pi\Theta\Sigma} &= \int \varepsilon^{\gamma\eta} \nabla_\gamma N^\Pi \nabla_\eta N^\Theta a^{\alpha\beta} \nabla_{\alpha\beta} N^\Sigma \sqrt{a} d\theta^1 d\theta^2, \\ K^{\Pi\Theta} &= \int a^{\gamma\eta} \nabla_{\gamma\eta} N^\Pi a^{\alpha\beta} \nabla_{\alpha\beta} N^\Theta \sqrt{a} d\theta^1 d\theta^2. \end{aligned} \qquad [6.15]$$

D and K are to be known as the damping and stiffness matrices, drawing an analogy with structural dynamics. C shall be referred to as the convection matrix.

The D and K transformations given in [6.14] can be written in equivalent matrix form. For instance

$$[D] = [S]^{\mathrm{T}} [D][S].$$

However this notation becomes problematic when dealing with the third order convection matrix, thus the tensor notion is to be adopted in the main.

6.4 Evaluation of the *D*, *C* and *K* matrices

To determine the values of the three matrices we require information about the slope and curvature for each element. The curvature appears in the convection matrix C and the stiffness matrix K. In order to evaluate this covariate derivative, we start with determining the surface base vectors.

Again we define the position vector $\mathbf{x}$ of an arbitrary point on the limit surface in terms of the primary unknowns,

$$\mathbf{x}(\theta^1,\theta^2)=N^{\Pi}(\theta^1,\theta^2)S_{\Pi}^{P}\mathbf{x}_P, \qquad [6.16]$$

Utilising the identities given in chapter two on differential geometry the surface base vectors can be defined simply as

$$\mathbf{a}_{\alpha}(\theta^1,\theta^2)=\mathbf{x}_{,\alpha}(\theta^1,\theta^2)=N^{\Pi}{}_{,\alpha}(\theta^1,\theta^2)S_{\Pi}^{P}\mathbf{x}_P, \qquad [6.17]$$

and subsequently we also have

$$\mathbf{a}_{\alpha,\beta}(\theta^1,\theta^2)=\mathbf{x}_{,\alpha\beta}(\theta^1,\theta^2)=N^{\Pi}{}_{,\alpha\beta}(\theta^1,\theta^2)S_{\Pi}^{P}\mathbf{x}_P. \qquad [6.18]$$

Taking values of

$$\theta^1=\frac{1}{3} \text{ and } \theta^2=\frac{1}{3},$$

and substituting in to the above equations gives values for $\mathbf{a}_{\alpha}$ and $\mathbf{a}_{\alpha,\beta}$ at the centre of the finite element (see Figure 6.5). These can form the basis of the one-point quadrature numerical integration. However it is worth noting that deriving the first and second derivatives numerically in this way means it would be a relatively simple procedure to implement a more complex quadrature evaluation.

Again using the relations set out in Table 3.2 the ingredients for evaluating the matrices can be obtained. From equation [6.17] we can calculate values for the covariant metric tensor as

$$a_{\alpha\beta}=\mathbf{a}_{\alpha}\cdot\mathbf{a}_{\beta}. \qquad [6.19]$$

The determinant can be evaluated from $a=a_{11}a_{22}-(a_{12})^2$.

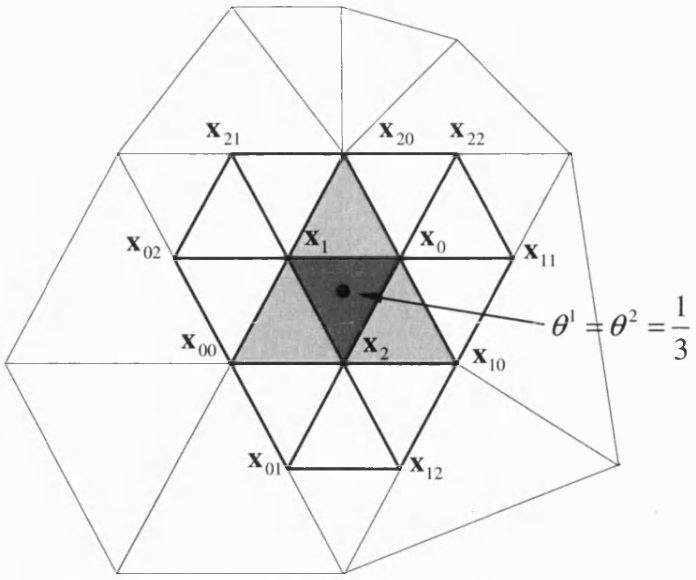


Figure 6.5 Evaluating properties of the limit surface at the centre of an arbitrary subdivision element

From the definition

$$a_{\alpha\mu}a^{\beta\mu} = \delta_{\alpha}^{\beta}.$$

the contravariant metric tensors can be found from standard matrix manipulation. In general

$$a^{\alpha\beta} = \frac{C^{\alpha\beta}}{a},$$

where $C^{\alpha\beta}$ is the cofactor associated with $a_{\alpha\beta}$. Hence, in two-dimensional space, we have the identities

$$a^{11} = \frac{a_{22}}{a},\ a^{22} = \frac{a_{11}}{a} \text{ and } a^{12} = \frac{-a_{12}}{a}.$$

The contravariant base vectors are now given by

$$\mathbf{a}^{\alpha} = a^{\alpha\beta}\mathbf{a}_{\beta}. \qquad [6.20]$$

The Christoffel symbols of the second kind are obtained by

$$\Gamma^{\eta}_{\alpha\beta} = \mathbf{a}^{\eta} \cdot \mathbf{a}_{\alpha,\beta}, \qquad [6.21]$$

and thus the second order covariant derivative for each element can be obtained

$$\nabla_{\alpha\beta} N^{\Pi} = N^{\Pi}{}_{,\alpha\beta} - \Gamma^{\lambda}_{\alpha\beta} N^{\Pi}{}_{,\lambda}. \qquad [6.22]$$

The first derivative of the scalar shape functions is simpler;

$$\nabla_{\alpha} N^{\Pi} = N^{\Pi}{}_{,\alpha}. \qquad [6.23]$$

Therefore using these quantities evaluated at the centre of the middle triangle the required integration simplifies and the local D, C and K matrices are defined by the relations

$$\begin{aligned} D^{\Pi\Theta} &= 2\sqrt{a}\, a^{\alpha\beta} \nabla_{\alpha} N^{\Pi} \nabla_{\beta} N^{\Theta}, \\ C^{\Pi\Theta\Sigma} &= 2\sqrt{a}\, \varepsilon^{\gamma\eta} \nabla_{\gamma} N^{\Pi} \nabla_{\eta} N^{\Theta} a^{\alpha\beta} \nabla_{\alpha\beta} N^{\Sigma}, \\ K^{\Pi\Theta} &= 2\sqrt{a}\, a^{\gamma\eta} \nabla_{\gamma\eta} N^{P} a^{\alpha\beta} \nabla_{\alpha\beta} N^{Q}. \end{aligned} \qquad [6.24]$$

As illustrated in Figure 6.5 the subdivided finite element consists of the central quarter of the original surface patch. The leading '2' appearing in [6.24] is therefore accounted for by the relation

$$Area = 4 \times \left(\frac{1}{2} \sqrt{a} \right).$$

6.5 Dynamic relaxation

The stream function formulation is non-linear in time. However equation [6.13] corresponding to a single instance in time, T, can be seen to be a Poisson's equation in $\dot{\psi}$,

$$(\dot{\psi}_J)_T D^{IJ} + (\psi_J \psi_K)_T C^{IJK} + \frac{\mu}{\rho}(\psi_J)_T K^{IJ} = 0. \qquad [6.25]$$

Thus, if we assume a known ψ, we can solve for $\dot{\psi}$. Once $\dot{\psi}$ has been calculated, time can be incremented and ψ updated with the process being reiterated for a new unknown $\dot{\psi}$. So the equation above is linear for each time step. Solving for $\dot{\psi}$ could therefore be carried out using a direct solution method in which the full damping matrix D is assembled and inverted. Here however an iterative form finding method is adopted. The problem of fluid flow requires a fine grid in order to model the areas of interest, thus the number of elements in the matrix is potentially very large. Most of these entries would in fact be zero due to the localised nature of any finite element method. To avoid formulating and storing this whole matrix, the solution at each time step will be found by dynamic relaxation. This method has the additional advantage that after the first few initial time steps, the solution will have already converged on a solution very close to that of each subsequent time step and so few iterations will be required.

Form finding as a process for determining minimum energy equilibrium geometries has a precedence in various physical experiments. Architects and Engineers such as Antonio Gaudí and Frei Otto used natural processes to define geometry for various structural forms. Some examples are given in the introduction to *Biomimesis and the geometric definition of shell structures in architecture*[3] included at the end of this thesis. Soap film surfaces are an interesting naturally forming geometrical example, which provide an analogy for a number of problems in mathematics and engineering. The surface described is one of constant tension, a minimal surface (that of minimum surface area or zero mean curvature). As such, soap film surfaces for small deflections form an analogy with any Laplacian type problem, examples of which being the torsion of

a prismatic bar[126] or the Poisson's equation [6.25]. A Laplace's equation is a Poisson equation equated to zero. Otto[138] was interested in natural forms as they represent this minimal geometry. They are inevitably efficient, with nature driven by the process of natural selection and minimum energy states[139].

Interestingly the idea of dynamic relaxation as a computational process was first conceived by Day[140] by drawing analogies between fluids and damped structural motion. It has since developed as a technique for form-finding and analysis of non-linear structural problems.

The principle, as proposed by Day, is to trace the damped motion of the structure over a period of small time increments until the structure finds static equilibrium. The theory is given in detail by Barnes[141] for the design and analysis of lightweight structures. Here a brief overview is given, relevant for the solution of the stream function formulation.

A finite difference form of Newton's second law of motion is used to determine the velocity $\mathbf{V}_i$ of each node of mass m_i in the system at each iteration of the process.

$$(\mathbf{F}_i)_t = m_i\left(\frac{(\mathbf{V}_i)_{t+\delta t/2} - (\mathbf{V}_i)_{t-\delta t/2}}{\delta t}\right). \qquad [6.26]$$

This gives

$$(\mathbf{V}_i)_{t+\delta t/2} = (\mathbf{V}_i)_{t-\delta t/2} + \frac{\delta t}{m_i}(\mathbf{F}_i)_t. \qquad [6.27]$$

The geometry of the system can thus be updated

$$(\mathbf{x}_i)_{t+\delta t} = (\mathbf{x}_i)_t + \delta t(\mathbf{V}_i)_{t+\delta t/2}. \qquad [6.28]$$

This new configuration results in new forces being generated at the nodes in the system, and thus the process can be repeated until a satisfactory low level of kinetic energy is in the system. This process is known as the Verlet algorithm.

To ensure stability and enable convergence viscous damping factors are applied to the system. Equation [6.27] may therefore look something like

$$(\mathbf{V}_i)_{t+\delta t/2} = \xi(\mathbf{V}_i)_{t-\delta t/2} + \zeta \frac{\delta t}{m_i}(\mathbf{F}_i)_t . \qquad [6.29]$$

For a form finding process in which the true motion of the structure is not being modelled and one is solely interested in convergence, the masses m_i are fictitious and all the factors can be chosen to optimise the rate of convergence. Here one wants the system to be just less than critically damped for optimum convergence.

Figure 6.6 over the page shows graphically the process of convergence. These images were produced performing dynamic relaxation on a rectangular mesh forming a simple mass-spring network. Similar implementations can produce convincing fabric behaviour with the addition of shear springs and self intersection algorithms[17, 69, 70].

Any reasonable starting geometry has no effect on the final equilibrium position. The word reasonable is used, as for the example given in Figure 6.6, which is highly geometrically nonlinear, starting geometries can easily be conceived in which the structure could become irreparably tangled and convergence on a sensible equilibrium geometry would be impossible.

For the current application where we are solving a linear system this is less of a problem, in addition, with the exception of the first time step, the *initial condition* of the relaxation will be the result of the previous convergence. Consideration of initial conditions shall be made with regard to individual test cases.

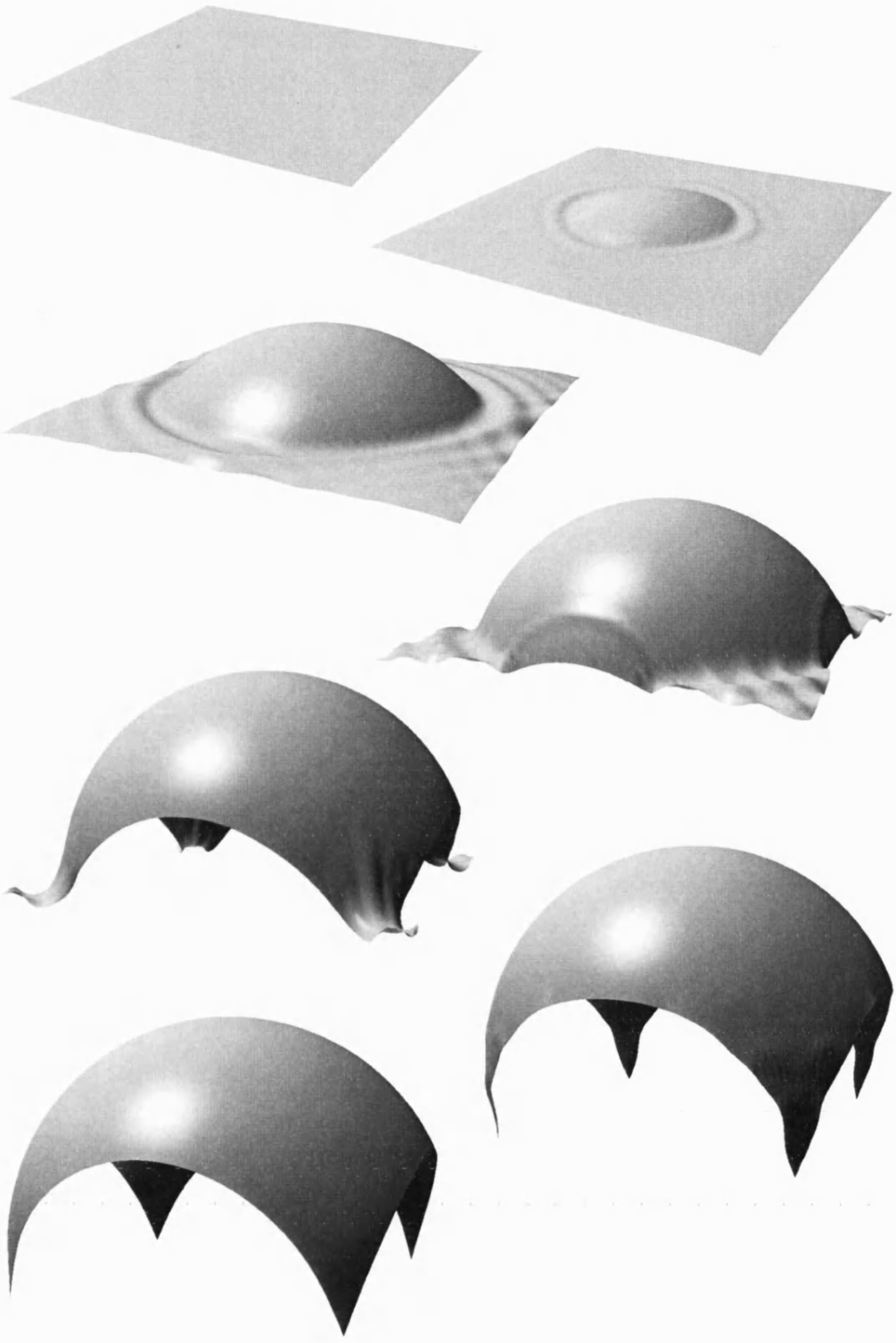


Figure 6.6 Dynamic relaxation of an initially planar geometry to a spherical surface. Images shown every 50 iterations

6.6 Conclusions

A novel approach to discretising the fourth-order stream function equation using an arbitrary topology mesh has been successfully derived.

The weighted residual form of the stream function has been written. This descretised form was then expressed using a subdivision shape function applicable to arbitrary meshes. The ability to define the required C^1 continuity was achieved by exploiting topological subdivision to create a patch of regular mesh which can then be interrogated using twelve regular shape functions. A generic subdivision matrix was devised in order to relate the limit properties of the continuum defined by twelve regular control points to the global degrees of freedom forming the arbitrary mesh. This work has satisfied points five and six of the thesis' core objectives.

Dynamic relaxation was introduced as a solution procedure. A direct solution method could be adopted. However the nature of the finite element method means the matrices formed shall be sparse and an iterative solution procedure avoids formulating and storing this information. These methodologies shall now be applied to solve the differential equations.

Before considering the highly non-linear test case of high Reynolds number flow past a circular cylinder, the implementation of the subdivision finite element shall be validated on the simpler linear test case of Stokes flow and plate bending. Following this, in chapter eight, a full non-linear solution procedure shall be derived.

Chapter seven

Validation of the subdivision element using a Stokes flow and plate bending formulation

7.0 Introduction

The aim of this chapter is to validate the arbitrary topology subdivision finite element derived in chapter six. This is carried out using three linear test cases solving the biharmonic equation. These benchmarks consist of both infinitesimal plate theory and Stokes flow models though analogy.

To achieve this purpose subdivision finite element boundary conditions sufficient for the application of plate bending are defined and a solution procedure utilising dynamic relaxation is set out.

By stating Reynolds number $Re << 1$ the stream function equation [6.13] simplifies to the descretised form of the biharmonic equation

$$\psi_J K^{IJ} = 0, \qquad [7.1]$$

where K^{IJ} is defined in equations [6.14] and [6.24].

Through analogy this also constitutes the equation for linear elastic plate flexure. The accuracy of the arbitrary topology finite element methodology can therefore be verified on linear elasticity problems before modelling the technically more challenging non-linear fluid behaviour.

7.1 Plate bending boundary conditions

At the boundary of a biharmonic function one can impose two conditions, as the problem essentially consists of two potential functions. The three classes of boundary condition familiar in the design of structures and plate theory are the clamped, pinned and free edge. These constitute impositions on the slope and deflection of the surface at the boundary. For example

clamped condition: $\dfrac{\partial \psi}{\partial \theta^1} = \dfrac{\partial \psi}{\partial \theta^2}$ = constant,

pinned condition: ψ = constant,

free edge: no constraints.

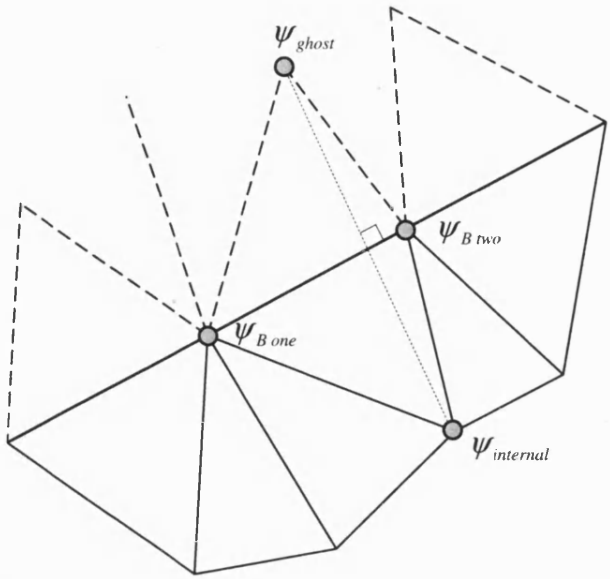


Figure 7.1 Ghost nodes formed at the boundary of the domain

The degrees of freedom controlling one subdivision finite element are not all localised within the patch, thus to apply conditions at a boundary we require the existence of *ghost nodes* outside the bounds of the domain. Similar to Cirak *et al.*[84] and Green and Turkiyyah[91], the plan geometry of the ghost nodes is created by mirroring the elements abutting the boundary across that boundary (as in Figure 7.1). The transverse position of the ghost node ψ_{ghost} depends on the condition being applied. The boundary conditions listed above will be considered in order.

7.1.1 Clamped boundary

Cirak *et al.*[84] say, for a clamped boundary condition

$$\psi_{ghost} = \psi_{B\,one} = \psi_{B\,two} = \psi_{internal} = 0.$$

Imposing zero at the boundary, the ghost node and its internal mirrored counterpart in this way over specifies the boundary. Cirak *et al.* uses a direct solution method, formulating the full stiffness matrix and then subsequently inverting it. The boundary conditions are applied using a penalty method, applying a very large stiffness (one hundred times the maximum diagonal

stiffness) to the nodes which are to be fixed. This in effect excludes that degree of freedom from the analysis. This is a method of avoiding clumsy renumbering that would otherwise be required if a degree of freedom was *actually removed* in a matrix method. In a relaxation method such as the implementation used here, degrees of freedom can be excluded simply by not including them in the relaxation iteration. This naturally fixes the position of that node.

Excluding the degrees of freedom at the row of nodes just inside the boundary blurs where the *actual boundary* is. For this implementation all internal nodes shall be included in the relaxation. The necessary position of the ghost nodes can be determined in alternative ways.

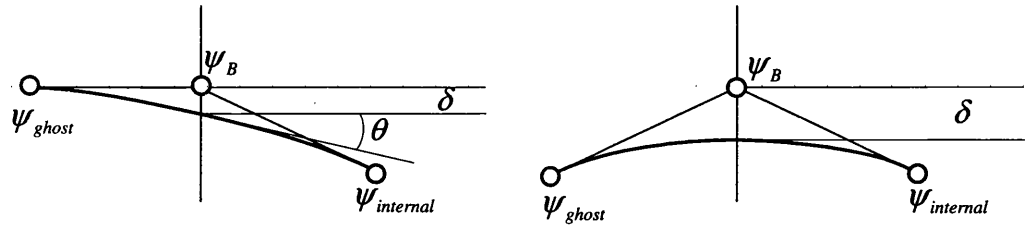
Allowing the internal nodes to move, but holding both boundary nodes and ghost nodes allows a rotation at the boundary.

$$\psi_{ghost} = \psi_{B\,one} = \psi_{B\,two} = 0,\ \psi_{internal} = \text{free}. \qquad [7.2]$$

This rotation is illustrated in Figure 7.2(a).

Secondly ghost nodes can be given the ψ displacement of their internal mirror counterparts, with the boundary nodes themselves held in place.

$$\psi_{B\,one} = \psi_{B\,two} = 0,\ \psi_{ghost} = \psi_{internal}. \qquad [7.3]$$



(a) (b)

Figure 7.2 Methods of imposing fixed boundary condition showing error in displacement δ and rotation θ. Fixed value of ghost node (a), ghost associated with internal node (b)

This gives a closer approximation to horizontal tangency at the boundary however a larger vertical displacement of the limit surface at the boundary is experienced due to the approximating, rather than interpolating, nature of the Loop subdivision method. Figure 7.2 shows the surface not passing through the nodes.

A third option is therefore to perhaps set a series of linear constraints such that the *actual surface* at the boundary satisfies both the conditions of horizontal tangency and prescribed displacement. This might be an over zealous approach to applying the boundary conditions. Any discretisation method is intrinsically an approximation, such that errors are integral to the implementation. Errors can be highlighted throughout the analysis, the resolution of the mesh, the method of integration over the finite element *et cetera*. An efficient implementation minimises these errors where possible. Therefore a holistic approach must be taken and concentrating on one aspect may be ineffective. One should choose a method which converges on the correct solution under successively refined meshes. Experiments in applying these fixed boundary conditions shall be considered in the following benchmark test cases.

7.1.2 Pinned boundary

To apply the pinned boundary conditions we do employ a similar methodology to Cirak *et al.*[84] For a symmetrical mesh at the boundary

$$\psi_{ghost} = \psi_{B\,one} + \psi_{B\,two} - \psi_{internal}\,.$$

For the case of an irregular boundary this can be generalised simply by projecting perpendicularly across the boundary as in Figure 7.1.

$$\psi_{ghost} = \beta_{one}\psi_{B\,one} + \beta_{two}\psi_{B\,two} - \psi_{internal}\,, \qquad [7.4]$$

where β_{one} and β_{two} are constants dictated by the plan geometry of the mesh.

7.1.3 Free edge

If the boundary nodes are also included in the relaxation loop, equation [7.4] constitutes a free edge.

7.2 Plate bending solution procedure

Utilising an iterative relaxation approach the following algorithmic structure is devised.

initialisation of analysis

- o generate base mesh
- o calculate K matrix
- o set initial conditions
- o assign solution constants $\frac{p}{D}$ and δt
- o calculate F_I, the $\frac{p}{D}$ value specific to each node

relaxation loop δt

- o calculate out of balance nodal forces

$$(f_I)_t = (\psi_J)_t K^{IJ} - F_I$$

- o relax ψ

$$(\dot{\psi}_I)_{t+\delta t/2} = \xi(\dot{\psi}_I)_{t-\delta t/2} - \zeta \frac{(f_I)_t}{K_{II}} \delta t, \text{ then}$$

$$(\psi_I)_{t+\delta t} = (\psi_I)_t + (\dot{\psi}_I)_{t+\delta t/2} \delta t$$

if ψ not sufficiently converged,

- O reapply boundary conditions and
- O reiterate

Figure 7.3 Summary of plate-bending algorithm structure

7.3 Benchmark one: Simply supported rectangular plate

The simply supported plate under uniform load, as also employed by Cirak *et al.*[84], is a useful simple test case to initially validate the conformity of the subdivision shape functions. The central deflection of a rectangular plate can be calculated using Navier's method as given in detail in Appendix C. Thus this test case can easily be verified against a known solution.

Hence for a rectangular plate of width a and breadth b, and with load p and stiffness D, choosing the values

$$\frac{p}{D} = 10^{-10} \text{ and}$$
$$a = b = 1000,$$

the Navier solution gives a central deflection of

$$\psi_{\max} = 0.406.$$

The error given by this implementation for progressively refined meshes can therefore be evaluated relative to the actual analytic solution given above. Figure 7.4 shows two characteristic meshes. The ghost nodes, not shown, are generated as detailed above.

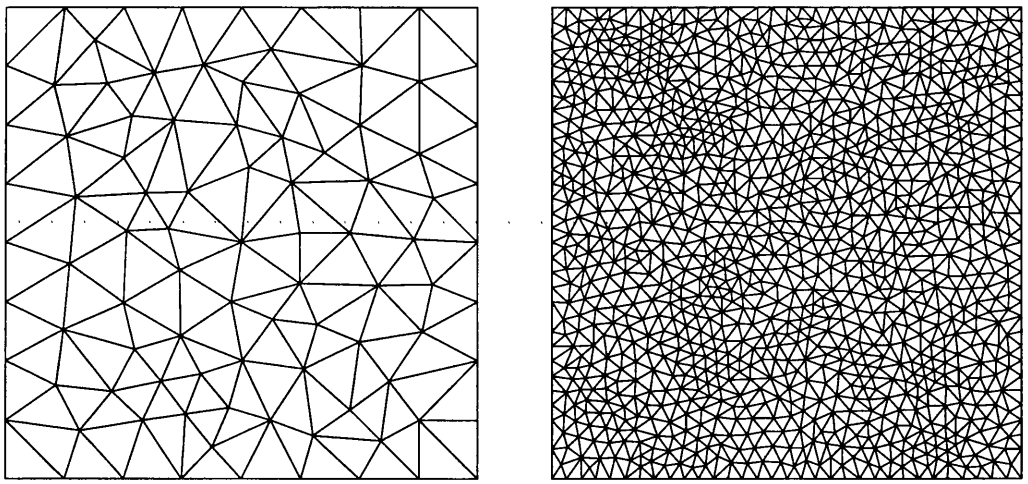


Figure 7.4 Typical arbitrary topology meshes used in this implementation. Pictured are meshes of 95 and 1132 degrees of freedom respectively

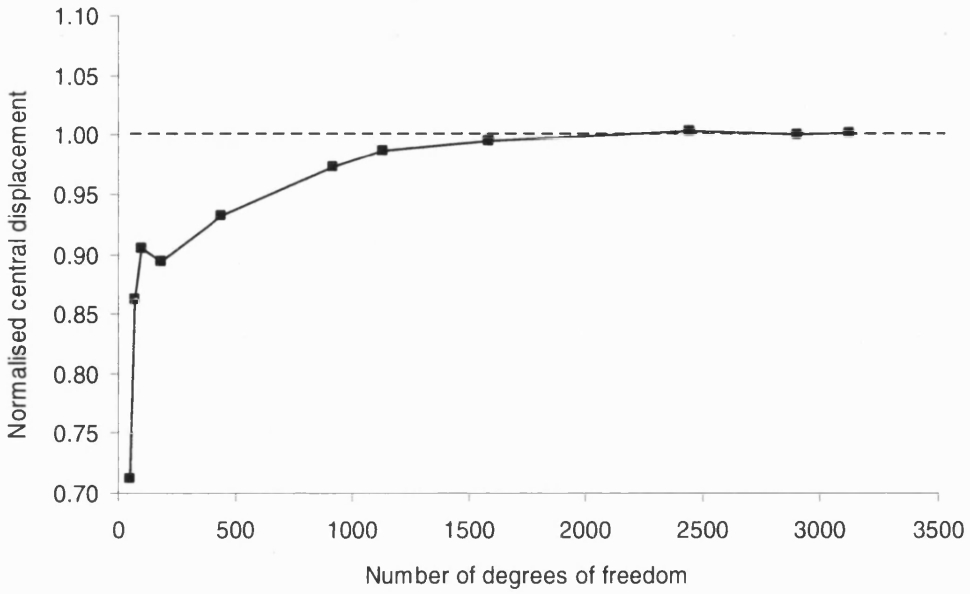


Figure 7.5 Convergence of the solution for the simply supported rectangular plate

Figure 7.5 illustrates the convergence on the correct solution (shown in Figure 7.6), with reasonable rapidity, thus proving the usefulness of this method. The results shown are for a completely irregular mesh using one-point quadrature. Following Cirak *et al.*[84] it is clear that this is adequate. The benefit of this approach is its simplicity of implementation, although further convergence tests on different quadrature methods may be useful to optimise performance of this approach.

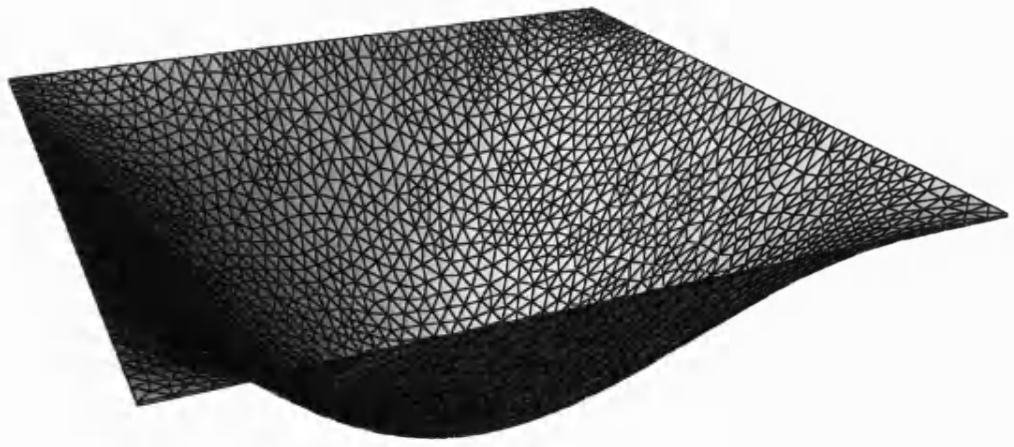


Figure 7.6 Deflected form for simply supported square plate under uniform loading (magnification factor applied to deflection)

Here we have used a completely arbitrary mesh as shown in Figures 7.4 and 7.6. Figure 7.5 shows slight oscillations in the convergence, seen particularly with small numbers of degrees of freedom. These are caused by this irregularity in the meshes used. The meshes of Cirak *et al.*[84] are well ordered with few extraordinary nodes. The finer meshes are generated by successive regular refinements isolating the irregular patches and further increasing the mesh's uniformity. Convergence is thus faster still on a series of similarly refined regular structured meshes. This is in agreement with what one would expect. On a regular patch of mesh C^2 continuity is achieved. Here we are concerned, however with completely arbitrary meshes as generated by a Delaunay triangulator. This gives flexibility in the boundary geometry and is very simple to implement. Mesh generation for complex geometry domains is hard work, thus the fewer constraints on mesh topology and mesh geometry the better.

Speed of convergence can also be increased in a comparatively straight forward way by distributing the nodes of the mesh in a more instructed way. Increasing the number of nodes in the boundary regions or where there is increased curvature of the surface. This is exploited to great use in the following test cases. This method of grid generation highlights the necessity for arbitrary topology meshes. Far away from the boundary few elements are required, near the boundary an increased concentration is required. Therefore at some point regularity must be broken.

Optimising the topology of the mesh through minimising the extraordinary nodes is beyond the scope of this thesis. These possibilities of further developments to the implementation shall be discussed in the recommendations of chapter ten.

7.4 Benchmark two: Laminar channel flow

As previously emphasised, there are few analytic solutions for fluid flow. An example of such a problem is laminar flow in a channel which is therefore a useful test case to validate against. The analogous structural system for this situation is a clamped rectangular plate subjected to differential settlement of the long edges.

Assuming the no-slip condition at the channel walls, the velocity profile v_x across the width of a rectangular channel is parabolic, varying from zero at the walls, to a maximum v_{max} in the centre of the channel. The derivation of this is given in Appendix D, the result being

$$v_x = v_{max}\left(1-\frac{y^2}{b^2}\right), \qquad [7.5]$$

and

$$\psi = v_{max}\left(y-\frac{y^3}{3b^2}\right). \qquad [7.6]$$

The notation is illustrated in Figure 7.7 below.

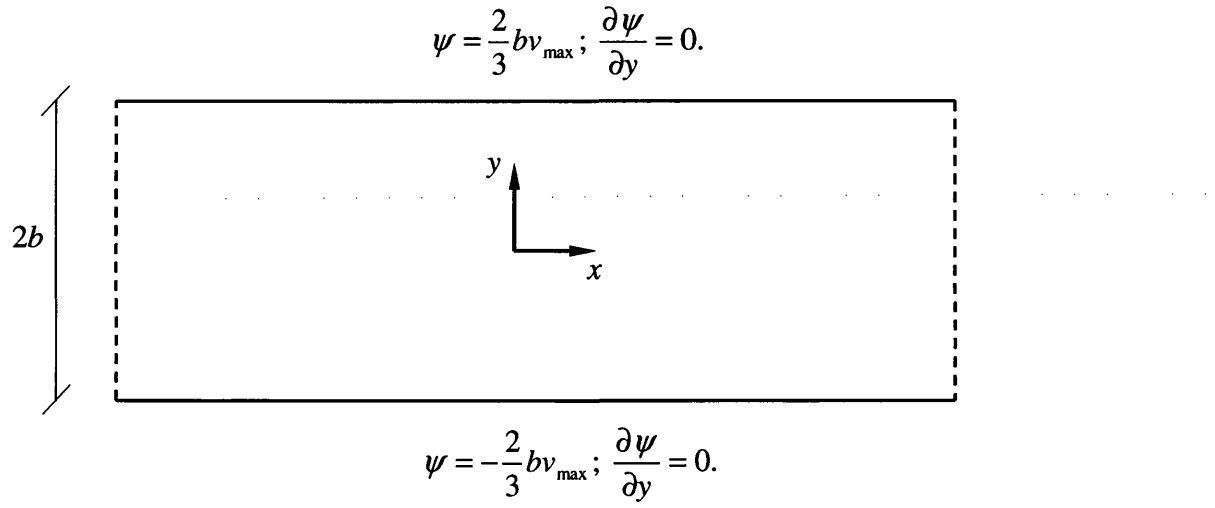


Figure 7.7 Geometry and imposed boundary conditions for channel flow

The graph below shows this discretisation method in comparison to the analytic solution. The full domain was modelled, although as the form is antisymmetric, only half of the channel is shown in the graph. The no-slip boundary condition is imposed using the two methods discussed in section 7.1. Identical meshes were used for each test, the only difference lying in the algorithm.

The analysis was performed with the ghost nodes held in place, equal to the boundary nodes, and then repeated with the ghosts' values changing in response to their internal counterparts. The graph highlights the performance of the different methods. The rotation occurring for the static ghosts can clearly be observed. On identical meshes the updated ghosts therefore clearly give a more accurate result than the static ghosts when compared to the analytic solution. The choice, therefore, becomes; is computational time best spent on updating the ghost nodes or with a larger number of fixed ghosts on a finer mesh? In certain cases the decision to maximise efficiency might not be so clear. However this implementation will later require the third order convection matrix C, as such minimising the number of elements is paramount. Therefore the no-slip boundary condition shall be imposed by associating the ghost nodes with their internal mirrored neighbours.

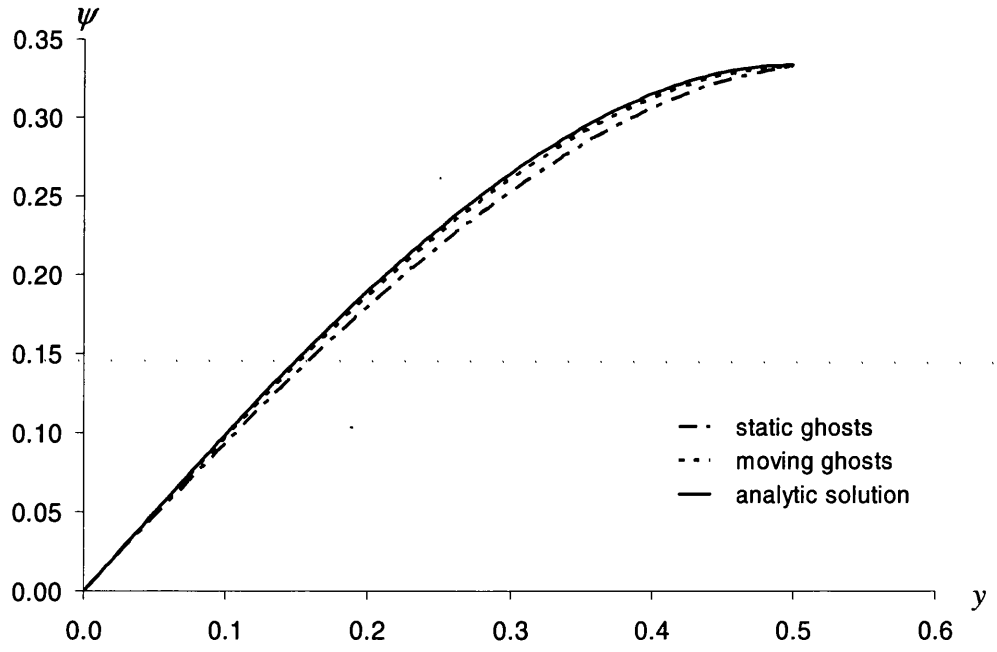


Figure 7.8 Performance of different methods of imposing the non-slip boundary condition on simple channel flow

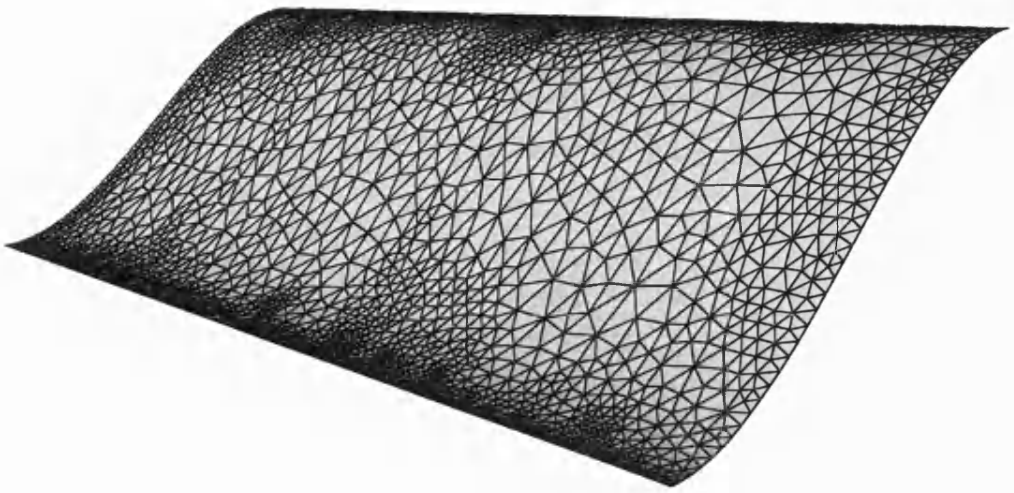


Figure 7.9 Stream surface for uniform channel flow

Rendering the stream function as a surface with ψ forming the third coordinate in three-dimensional space, as pictured in Figure 7.9, allows the plate bending and viscous fluid analogy to be readily visualised. Notice in this figure the refinement of the mesh in the boundary layer and coarser mesh elements in the faster flowing central region where the surface is flatter. The stream lines below are the contours of the surface in Figure 7.9 viewed in plan. They clearly illustrate this flow geometry. In the centre of the channel the lines are close together showing fast flowing fluid, with the lines farther apart towards the boundary where flow is slow.

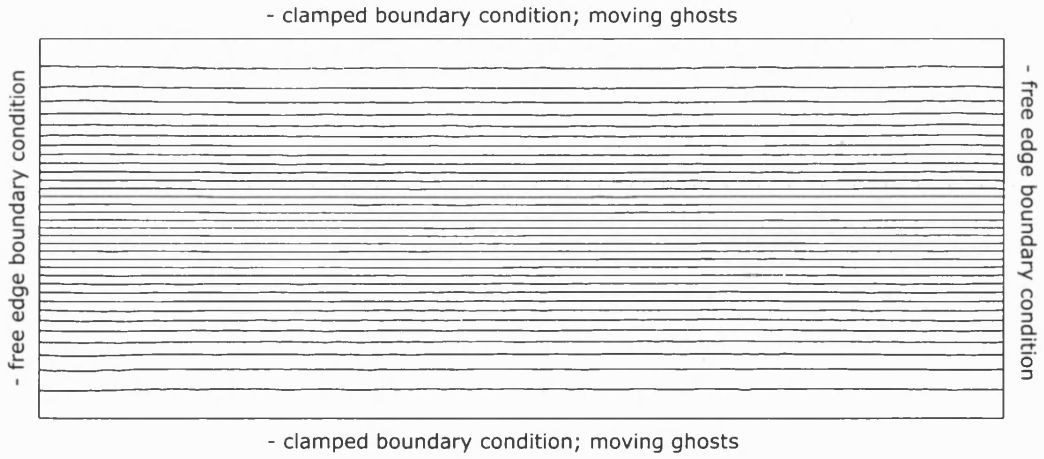


Figure 7.10 Stream lines for uniform channel flow. Stream line separation 0.020; width 1.0

7.5 Benchmark three: Stokes' flow past a circular cylinder

This is the last test case considered in which we will not be relying on other numerical and physical experimentations for validation. Stokes flow, or creeping flow corresponds to phenomena where inertial forces are negligible with regard to the viscous component. Here the Reynolds number

$$Re = \frac{\rho V d}{\mu} << 1.$$

The case of two dimensional creeping flow past a cylinder is an interesting one as for a finite diameter cylinder in an infinite expanse of fluid it proves to have no solution[142]. If we consider the analogous plate bending problem, illustrated below, this paradox is resolved.

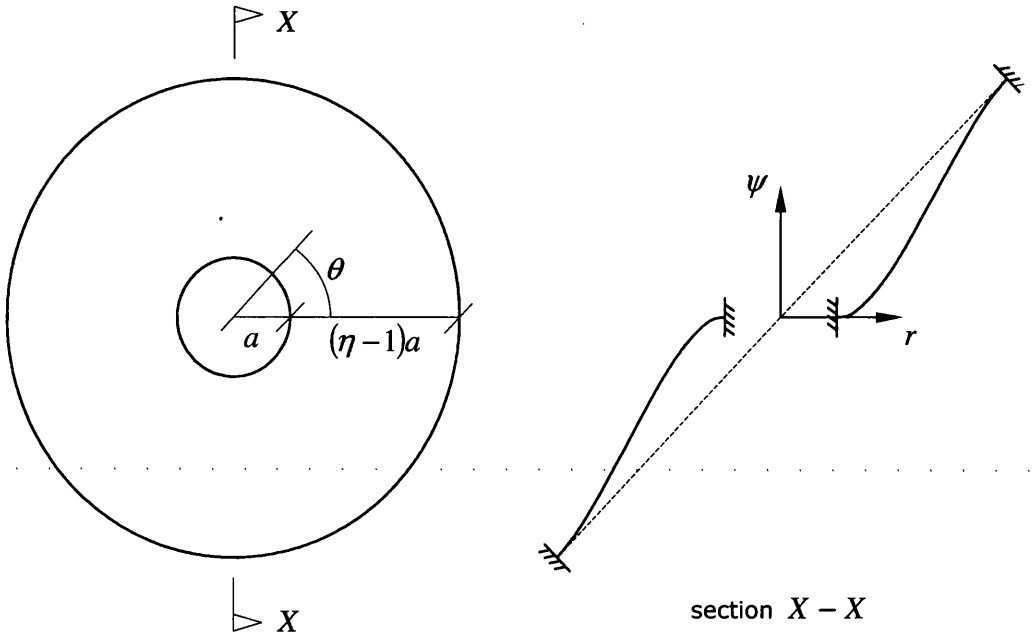


Figure 7.11 Geometry of circular plate under bending. Section illustrates clamped boundary conditions applied to annulus

The case of a circular plate with a concentric circular hole under bending has an analytic solution, the full derivation of which can be found in Appendix E. The final result is

$$\psi = \frac{a\left(\frac{1}{2}\left(\frac{r^3}{a^3} - 2\frac{r}{a} + \frac{a}{r}\right) - \left(\eta^2 + 1\right)\left(\frac{r}{a}\ln\frac{r}{a} - \frac{1}{2}\frac{r}{a} + \frac{1}{2}\frac{a}{r}\right)\right)}{\left(\eta^2 - 1\right) - \left(\eta^2 + 1\right)\ln\eta}\cos\theta. \qquad [7.7]$$

Figure 7.12 shows a good convergence on the analytic solution for $a = 0.5$ and $\eta = 4$, using progressively finer meshes. Conformance to the tangency boundary condition around the circumference of the circular hole can be seen.

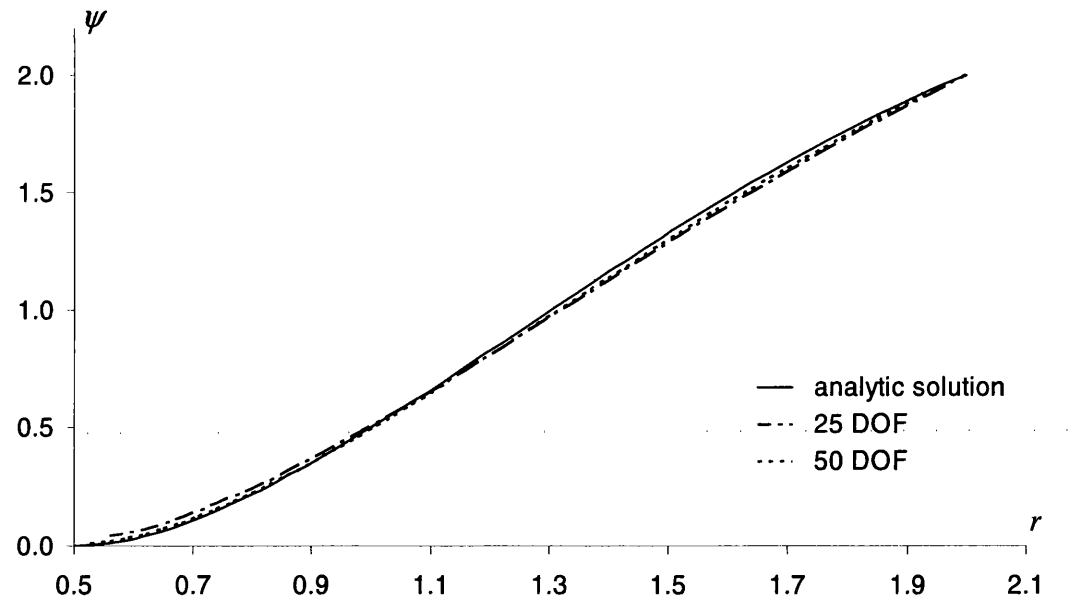


Figure 7.12 Comparison of the numerical method with the given analytic solution for a clamped circular plate with circular hole under bending. $a = 0.5$, $\eta = 4$. The graph shows the performance of two meshes with an approximate average of 25 and 50 degrees of freedom spanning the cross section

Figure 7.13 The ψ-surface for circular plate bending and creeping flow past a circular cylinder in a finite domain

The surface for circular plate bending and creeping flow past a circular cylinder in a finite domain is shown in Figure 7.13. Contours of this surface are given in Figure 7.14. Other methods of flow visualisation, as alternatives to the streamline, are streaklines and particle paths. Streaklines correspond to the patterns observed when experimentally introducing dye into the flow. Particle paths, for all but the simplest flow are incredibly complex and chaotic making their analysis difficult. For creeping flow all three cases are equivalent.

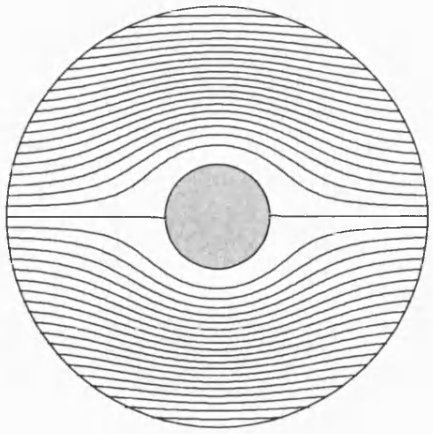


Figure 7.14 Circular plate bending and creeping flow past a circular cylinder in a finite domain. $a = 0.5,\ \eta = 4$, contour spacing 0.1

7.6 Conclusions

The three biharmonic test cases considered in this chapter have successfully enabled validation of the devised arbitrary subdivision element methodology, satisfying the eighth objective set out in chapter one. Obtaining C^1 continuity is the problematic part of the finite element implementation; the C^0 continuity required for the rest of the matrices could be achieved with planar triangles and simple geometric continuity. The arbitrary subdivision element was seen to perform well, converging on the solution efficiently under successive refinements of the meshes.

A dynamic relaxation solution procedure was devised avoiding the formation and multiplication of the full matrices. The algorithm proposed proved to be an efficient approach, well suited to the fine finite element meshes which shall be demanded in the non-linear fluid implementation of chapters eight and nine.

Approaches to the application of boundary conditions were also set out and their viability tested. The proposed relaxation methods demanded novel rules for imposing the boundary conditions to be developed. Different approaches to enforcing the no-slip condition through ghost nodes where tested. The

methodology of updating the ghost nodes was seen to be most effective and shall be adopted for the test cases in the following chapters.

This subdivision methodology has proven successful on this simple linear benchmarks. However in order to model more than the most trivial flows one must include the inertia terms in the implementation. In chapter eight the complexity of the relaxation algorithm shall therefore be increased to include a temporal nonlinearity into the program, affording simulation of more interesting fluid behaviour and surface geometries.

Chapter eight

Formulation of the stream function algorithm

8.0 Introduction

In this chapter the previously devised algorithm for solving the linear biharmonic equation is expanded to include the additional convective terms of the full temporally non-linear stream function equation.

Firstly an approach for imposing the differential equation boundary conditions for the special case of fluid flow past a cylinder is devised. In addition the initial condition for this test case shall be specified. The initial conditions for temporally non-linear problems are obviously more critical affecting the result as a static solution is not being converged to.

This chapter formulates the final tools required for the solution of the fourth-order stream function equation for flow past a circular cylinder.

8.1 Stream function boundary conditions

For the stream function we have analogous boundary constraints to the plate bending conditions described in chapter seven. For viscous flow past an obstacle the no-slip boundary condition is applied at the surface. The fluid at the interface must be stationary, therefore in terms of ψ we have

At a body with no-slip: $$\frac{\partial \psi}{\partial \theta^1} = \frac{\partial \psi}{\partial \theta^2} = 0,$$

which is equivalent to the horizontally clamped condition employed for plate bending.

In free stream: $$\frac{\partial \psi}{\partial x} = 0, \quad \frac{\partial \psi}{\partial y} = U_\infty .$$

Thus it is the slope of the stream function and not the absolute value which is the immediate constraint at the boundary of any solution. The absolute value of ψ is dependent on the test case, the particular boundary geometry and the chosen frame of reference. For the interesting case of flow past a bluff body which shall constitute the basis of the following chapter there is an additional condition that must be fulfilled allowing a solution for the absolute value to be found.

In chapter nine the case of fluid flow past a cylinder shall be considered for varying values of Reynolds number Re. For high Re the flow is asymmetric so that at the boundary of the cylinder the value of ψ relative to the rest of the domain will fluctuate as the volume of fluid flowing either side of the cylinder increases and decreases. The flow is only symmetrical at low Reynolds numbers where vortex shedding is not occurring.

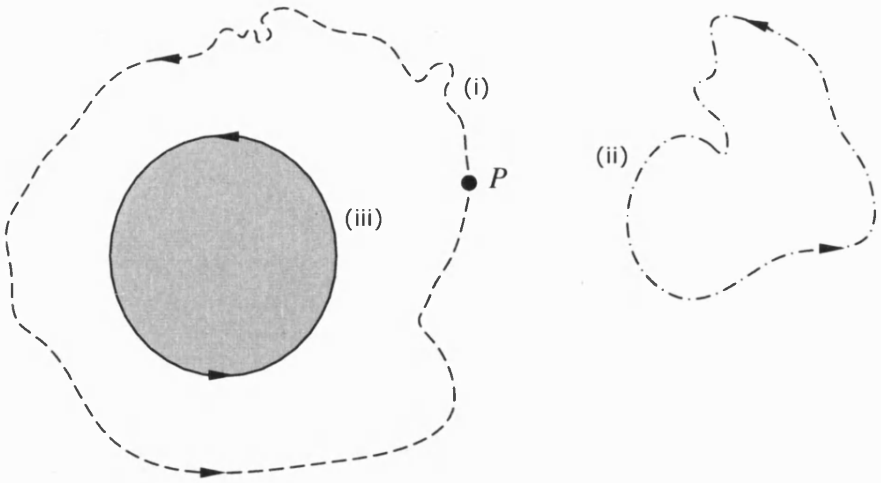


Figure 8.1 Closed paths in the fluid domain for flow past a cylinder. Path (iii) coincides with the cylinder boundary.

Figure 8.1 shows three arbitrary closed paths (i, ii & iii) traced out in the fluid for a given moment in time. The value of pressure at point P, at the start and end of the loop must be equal. In other words, integrating the pressure gradient around a closed path must give zero. Therefore revisiting the equations derived in section 4.3 we can say

$$\oint \varepsilon^{\alpha\beta}\left(\frac{\partial}{\partial t}+\varepsilon^{\lambda\eta}\nabla_{\eta}\psi\nabla_{\lambda}-\frac{\mu}{\rho}a^{\lambda\eta}\nabla_{\lambda\eta}\right)(\nabla_{\beta}\psi)d\theta^{\alpha}=-\oint\nabla_{\alpha}pd\theta^{\alpha}=0.$$

If we chose the special case of a path that coincides with the cylinder boundary (see path (iii) in Figure 8.1) the above equation simplifies. As $\nabla_{\alpha}\psi=0$ on the boundary, and we can say

$$\oint \varepsilon^{\alpha\beta}\frac{\mu}{\rho}\nabla_{\gamma\eta\beta}\psi d\theta^{\alpha}=0. \qquad [8.1]$$

This is a second condition to be applied at the boundary allowing the volume of fluid flowing either side of the cylinder to fluctuate as the vortices are shed periodically. It can be visualised as a statement of equilibrium of vertical shear forces due to plate bending around the cylinder.

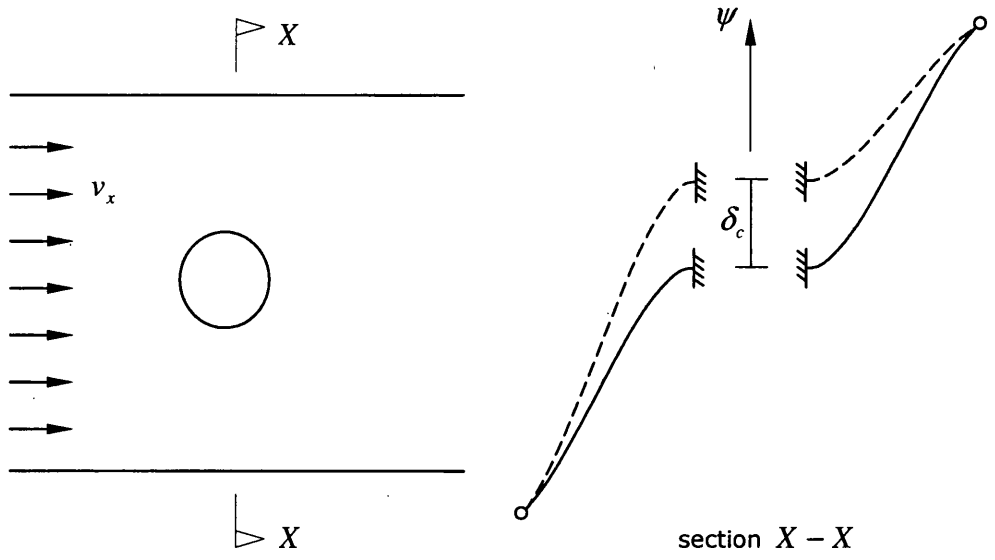


Figure 8.2 Section through the stream function limit surface illustrating the deflection of the cylinder boundary to maintain equilibrium

As illustrated in Figure 8.2, the transverse deflection of the boundary δ_c equals the sum of the nodal residual forces f_c round the cylinder boundary, multiplied by the stiffness k_c

$$\delta_c = f_c k_c .$$

The force-deflection relation is linear (our stream-function-structures analogy is based on small deflection theory), thus the constant value of the cylinder stiffness k_c can be evaluated using a trial displacement and evaluating the resultant out of balance force before the simulation is initiated.

For the current implementation the cylinder is actually fixed at zero and the rest of the domain, including all other boundaries, is moved to regain equilibrium. This causes the zeroth streamline to always coincide with the cylinder boundary enabling consistent comparison of results.

8.2 Stream function initial conditions

Before modelling this scenario with our code, the initial conditions for the relaxation must be determined. For a linear problem converging on a stationary solution, this is obviously not critical. However for the time dependant flows in later test cases it shall be important.

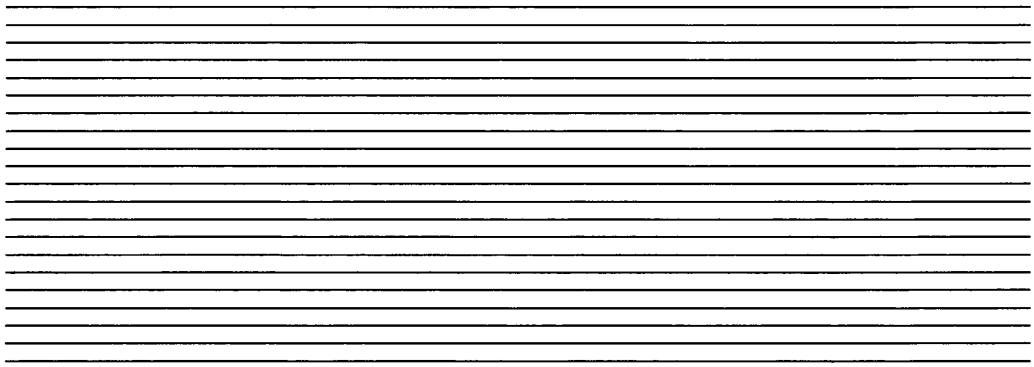
Here we introduce potential or inviscid flow. Potential flow past a cylinder has a finite value for v_θ at the boundary, violating the no-slip condition. Potential flow is probably close to the starting conditions for instantaneously moving a cylinder from rest in a fluid.

As Prandtl and Tietjens[147] state, the potential flow streamlines around an aerofoil can be calculated using a method of conformal mapping. There are various forms of mapping (see Struik[120] page 168), here we are concerned with preserving angles between metrics on the surface.

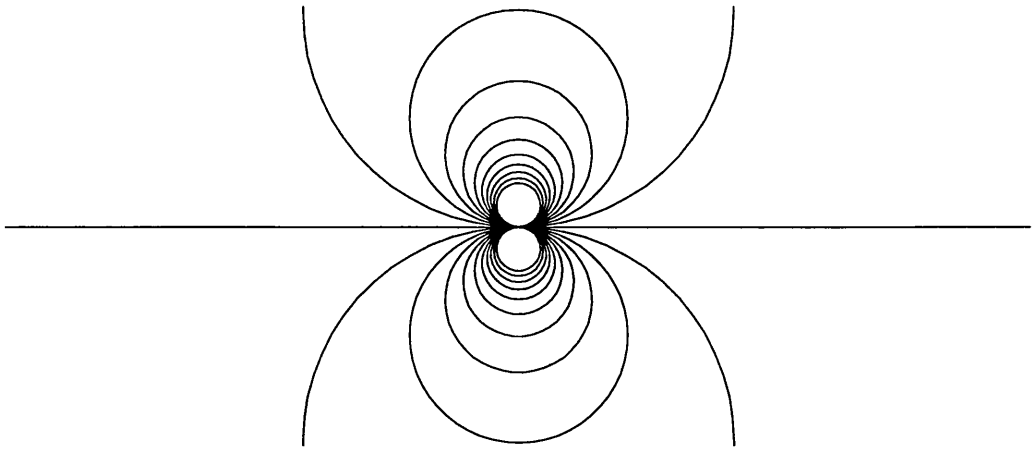
The equation below is devised from uniform flow superimposed with a dipole giving potential flow past a circular cylinder, see Figure 8.3

$$\begin{aligned} \phi + i\psi &= z + \frac{R^2}{z}, \\ &= x + iy + \frac{R^2}{x+iy}. \end{aligned} \qquad [8.2]$$

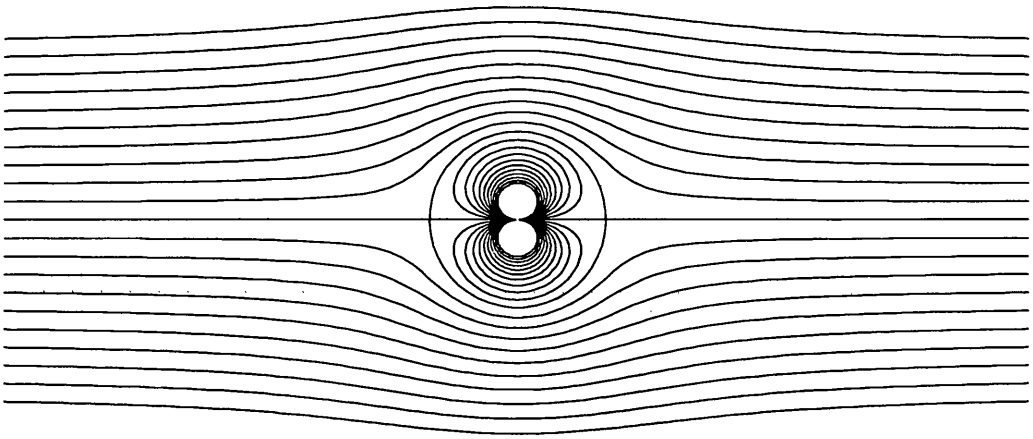
For a finite element where we are given x and y at the vertices of an element, to impose these initial conditions we need ψ in terms of these knowns.



(a)



(b)



(c)

Figure 8.3 Uniform flow (a) and dipole (b) superimposed to give potential flow streamlines for inviscid flow past a circular cylinder (c)

Therefore multiplying top and bottom of second term on the right hand side of equation [8.2] by the complex conjugate of z

$$\phi + i\psi = x + iy + \frac{R^2(x-iy)}{(x+iy)(x-iy)},$$
$$= x + iy + \frac{R^2(x-iy)}{x^2+y^2},$$

thus we have the stream function

$$\psi = y - \frac{R^2 y}{x^2+y^2},$$

and also the equipotentials

$$\phi = x + \frac{R^2 x}{x^2+y^2}.$$

Using this potential geometry as a starting condition, the Stokes flow solution was modelled. The initial conditions imposed at the outer boundary were uniform flow. The discrepancy caused by this was minimised and proved negligible due to the size of the domain.

8.3 Temporally non-linear stream function solution procedure

Recalling equation [6.25] from chapter six

$$(\dot{\psi}_J)_T D^{IJ} + (\psi_J \psi_K)_T C^{IJK} + \frac{\mu}{\rho}(\psi_J)_T K^{IJ} = 0. \qquad [6.25]$$

Using dynamic relaxation, the temporal discretisation can be thought of as consisting of two nested iterative loops. The stream function surface ψ is incremented in the main time step, ΔT. The form finding of $\dot{\psi}$ is carried out in subsidiary time steps δt.

$$\delta t = \frac{\Delta T}{n},$$

where n is the maximum number of subsidiary iterations. The average percentage change in $\dot{\psi}$ is used as a cut off for the relaxation. When this has reached a minimum threshold the relaxation for that time step terminates. The maximum number of subsidiary iterations n is imposed terminating the relaxation prematurely in the event of slow convergence or divergence.

A trace of the average percentage change in $\dot{\psi}$ over time will be of the form shown in Figure 8.4.

The initial few time steps can be seen to require a number of subsidiary iterations. However, the solution settles down after a while and few subsidiary time steps are required to converge. For small values of ΔT in fully developed flow, only one or two iterations of δt are required.

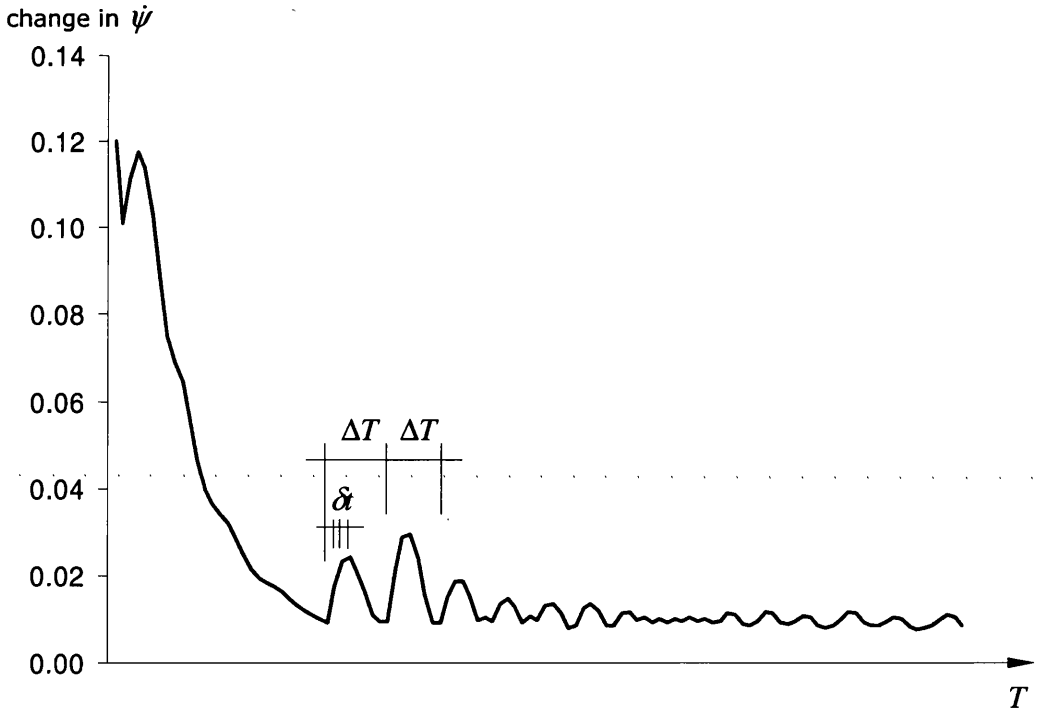


Figure 8.4 Example of time history graph showing convergence of $\dot{\psi}$ within each time step for initial stages of a simulation. Produced during Re = 100 flow past a cylinder presented in section 9.2

Within each main time step $\dot{\Delta T}$ the relaxation of $\dot{\psi}$ over subsidiary time steps δt can be considered as *virtual*, similar to a form finding process in structural design in which imaginary time steps, relaxation factors and masses are chosen arbitrarily in order to stabilise and speed up the convergence of the solution. Thus, for clarity, the identity

$$\dot{\psi}_I \equiv \eta_I,$$

is used to distinguish between the *real* $\dot{\psi}$ in equation [6.25] and the virtual relaxation process undertaken in the subsidiary time steps. This relaxation algorithm takes the following form.

Using the value of ψ from the previous time step T, and the current η at t, we can evaluate the *residual force* f_I at the I th internal mesh point as

$$(\eta_J)_t D^{IJ} + (\psi_J \psi_K)_T C^{IJK} + \frac{\mu}{\rho}(\psi_J)_T K^{IJ} = (f_I)_t. \qquad [8.3]$$

These forces constitute the error in the solution of the Poisson's equation, which is to be minimised. It can be noted that in equation [5.43] the second two terms only contain ψ_I, independent of time t and thus need not be recalculated for each iteration of the subsidiary time steps.

In a similar form to [6.29] we can say

$$(\dot{\eta}_I)_{t+\delta/2} = \xi(\dot{\eta}_I)_{t-\delta/2} - \zeta \frac{(f_I)_t}{D_{II}} \delta t, \ (I \text{ is not summed}) \qquad [8.4]$$

where ξ and ζ are relaxation factors. ξ is the dynamic relaxation carry over factor and ζ is the viscous damping factor.

As $\zeta \delta t^2 \rightarrow 1.0$ nodal points will tend to over shoot the correct solution causing oscillations and divergence.

Therefore in order to under-relax the solution the relationship between ζ and δt is governed by

$$\zeta \delta t^2 \approx 0.2 .$$

Again the value of ξ controls the speed of convergence, with oscillation and instability limiting its upper value $\xi \approx 0.9$, with this value increasing for high numbers of degrees of freedom. For simulations with tens of thousands of degrees of freedom ξ can exceed 0.99.

Here it is assumed that the damping matrix (if the whole matrix was to be formed) is diagonally dominant and so it is effective to move each node in proportion to the leading diagonals of the matrix. Therefore in implementing this algorithm the individual local stiffness matrices and the overall leading diagonal terms are stored.

η can now be updated from $\dot{\eta}$ by

$$(\eta_I)_{t+\delta t} = (\eta_I)_t + (\dot{\eta}_I)_{t+\delta t/2}\, \delta t . \qquad [8.5]$$

After n iterations, or with the residual forces sufficiently minimised to an acceptable level of error, ψ can be updated using a simple Eulerian method of incrementation

$$(\psi_I)_{T+\Delta T} = (\psi_I)_T + (\dot{\psi}_I)_T \Delta T, \qquad [8.6]$$

Note that $(\dot{\psi}_I)_T$ is used rather than $(\dot{\psi}_I)_{T+\Delta T/2}$. This is because it is calculated from $(\psi_I)_T$.

After updating ψ, the boundary conditions are reasserted before the process is reiterated. This algorithm is illustrated in Figure 8.5.

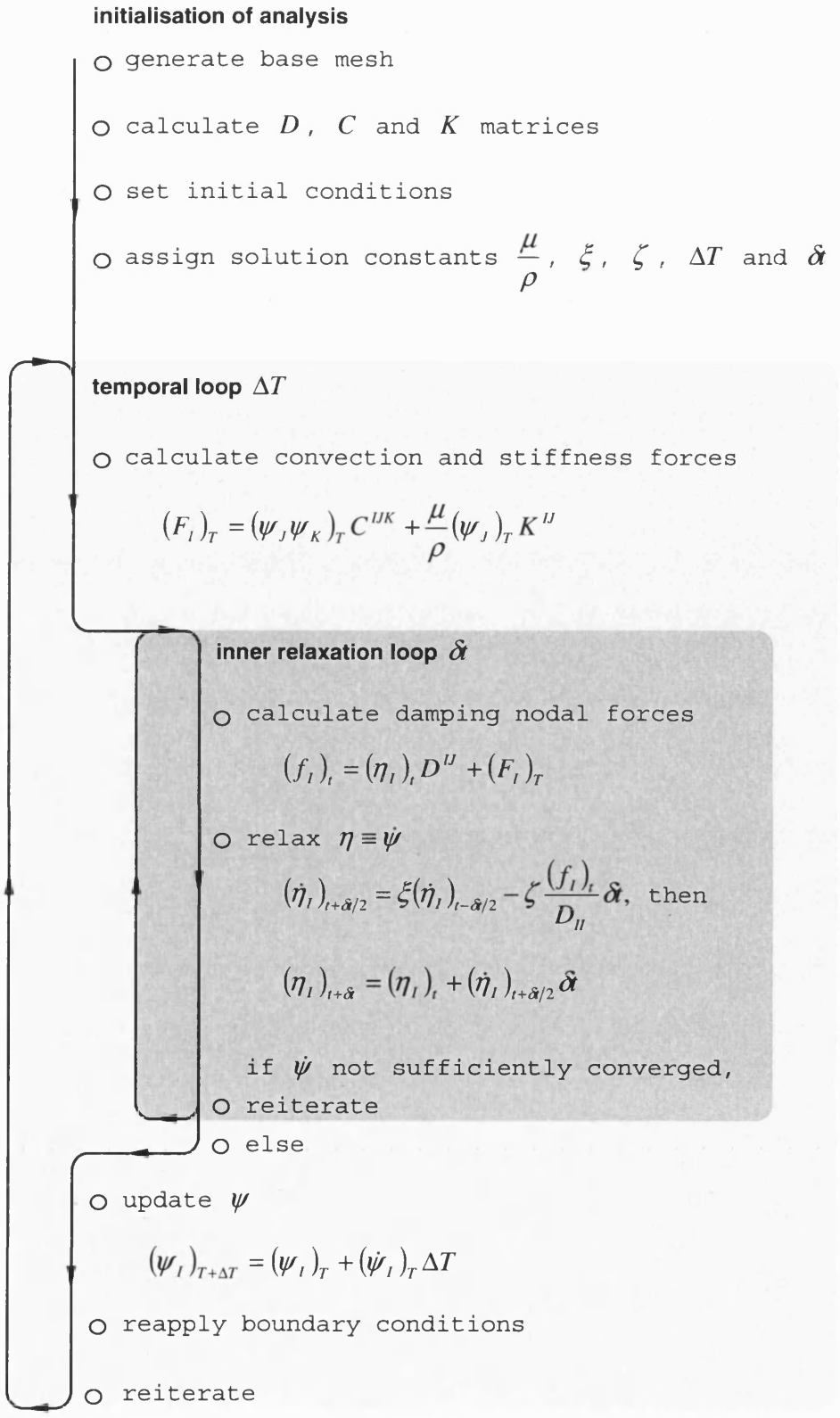


Figure 8.5 Summary of fluid-flow algorithm structure. See also Figure 7.3 for comparison with linear plate-bending algorithm

8.4 Conclusions

A novel approach for the application of the cylinder boundary conditions has been devised. This along with the successful definition of the required initial conditions satisfy objective nine of this thesis.

An approach for the solution of the non-linear stream function equation has been set out demonstrating objective ten. This new approach consists of a temporal discretisation, each iteration of which forms an iterative relaxation form finding the geometry of the Poisson's equation at each given time step.

In the previous chapter the subdivision finite element was successfully validated using the linear biharmonic equation. The new extended non-linear methodology is now applied to the case of non-trivial fluid flow past a circular cylinder. The accuracy of the approach shall be ascertained through validation of the results against existing experimental data.

Chapter nine

Application of the formulation to flow past a cylinder

9.0 Introduction

The problem forming the focus of this research and for validation of this implementation is viscous fluid flow past a circular cylinder. This is an interesting test case. From elementary boundary geometries and initial conditions intricate and complex flow patterns naturally develop as a result of the inherent instabilities in the Navier-Stokes equations. Different characteristic behaviour is experienced as the Reynolds number varies.

Flow past a cylinder also constitutes a valuable test case for understanding the behaviour of fluids as many of the important characteristic behaviours of fluid flow, such as vortex shedding and galloping can be observed during this test case.

Six cases of flow past a cylinder shall be considered with increasing complexity of flow for increasing Reynolds number. The accuracy of the simulations shall be verified against existing experimental results.

9.1 Flow past a circular cylinder *Re* = 40

As the Reynolds number is increased into the approximate region $4 \leq Re \leq 40$, the first bifurcation of the Navier-Stokes equations occurs. All other parameters being equal, increasing Re corresponds to a lower value for μ/ρ. The inertia terms therefore have increasing influence. In this range of Re they cause separation of the boundary layer to occur and a recirculation bubble forms on the leeward side of the cylinder. This is diagrammatically illustrated in Figure 9.1.

As the Reynolds number increases this vortex couple, called Föppl vortices, stretch and a relationship between Re and the recirculation length s/d can be observed[148].

The domain geometry for the problem is given in Figure 9.2

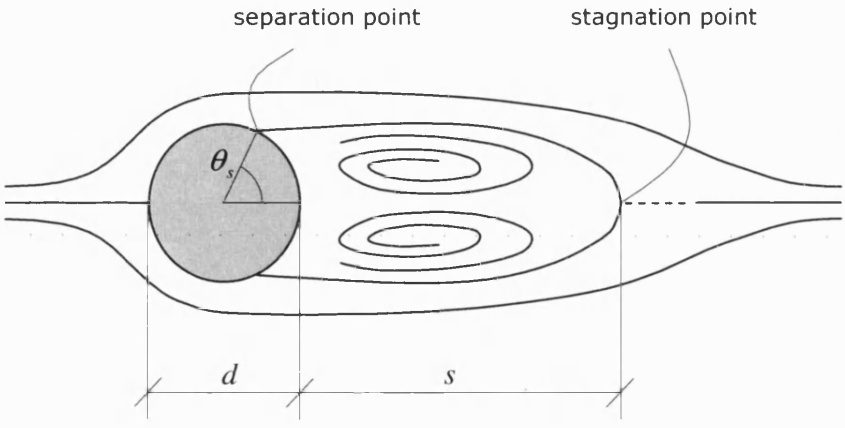


Figure 9.1 Geometry of the wake for flow past a cylinder at $Re < 46$, illustrating parameters, the separation angle θ_s and the recirculation length s/d

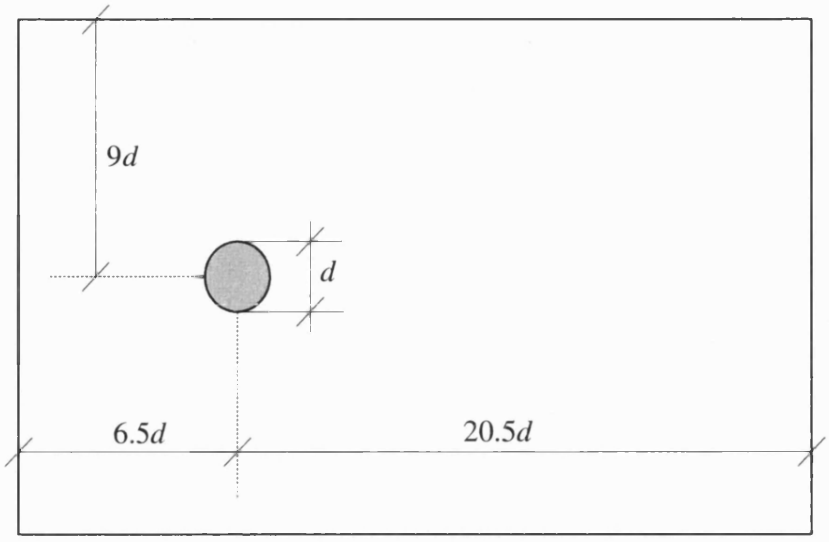


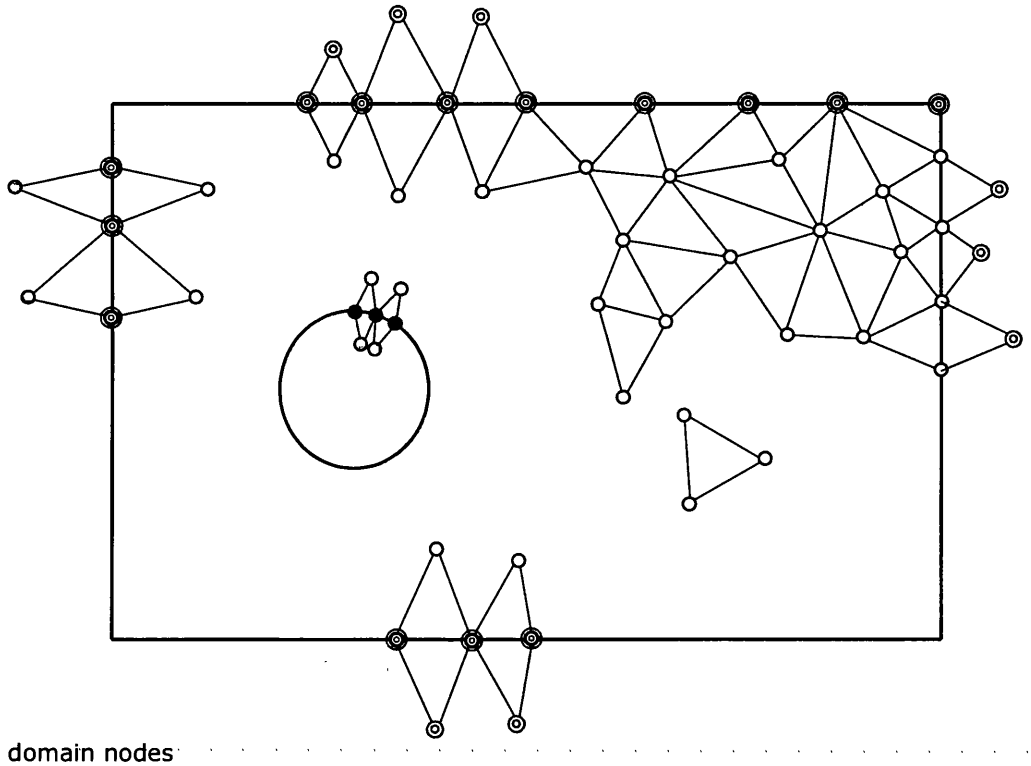
Figure 9.2 Relative dimensions of the fluid domain. (not to scale)

The boundary condition to be applied at the cylinder is obviously the no-slip/clamped constraint. However the choice of boundary conditions for the remaining four sides of the rectangular domain is not as immediately obvious. When considering these boundaries, it can again be useful to exploit the structural analogy.

As an objective we can attempt to replicate a cylinder introduced into an infinite flow of unit velocity. The left hand boundary can therefore be clamped at a gradient of unity parallel to the edge of the domain (whilst the surface gradient is horizontal perpendicular to the boundary). However the conditions applied to the leeward and horizontal boundaries ideally should not constrain the flow. It is of importance, particularly for the high Re flows later, that perturbations in the surface can be convected out of the domain. A free edge boundary is given for the right hand edge. However also allowing free boundaries on the top and bottom is not constrictive enough. Initial experiments with a free boundary showed that there proved to be a lack of containment to the flow, the instabilities caused by the introduction of the cylinder encourages the surface down stream of the cylinder to flatten out horizontally.

Freeing the boundary allows flow in and out of the domain. If one's aim is to model an infinite flow there would always be more fluid outside the boundary of the domain containing the flow. Thus this implementation uses a *pinned* boundary condition at the top and bottom. This constitutes a slip boundary. Containment of flow affecting the geometry of the wake is minimised by using a sufficiently large domain size in relation to the cylinder diameter, as illustrated by Figure 9.2. The boundary conditions used for flow past a cylinder are therefore summarised in Figure 9.3 below.

As illustrated in Figure 9.1, the geometry of concern forms in the near wake of the cylinder. Areas at a distance far from the cylinder are of little relative interest.



domain nodes
- o node type 0. included in relaxation loop*
- ◎ node type 1. excluded from relaxation*
- ● node type 2. fixed

ghost nodes
- o node type 3. ψ value equal to internal node*
- ◎ node type 4. ψ value mirrored across boundary*

* node types 0, 1, 3 and 4 are updated to equalise transverse shear on the cylinder

Figure 9.3 Boundary conditions for flow past a cylinder, imposed through different mesh node types

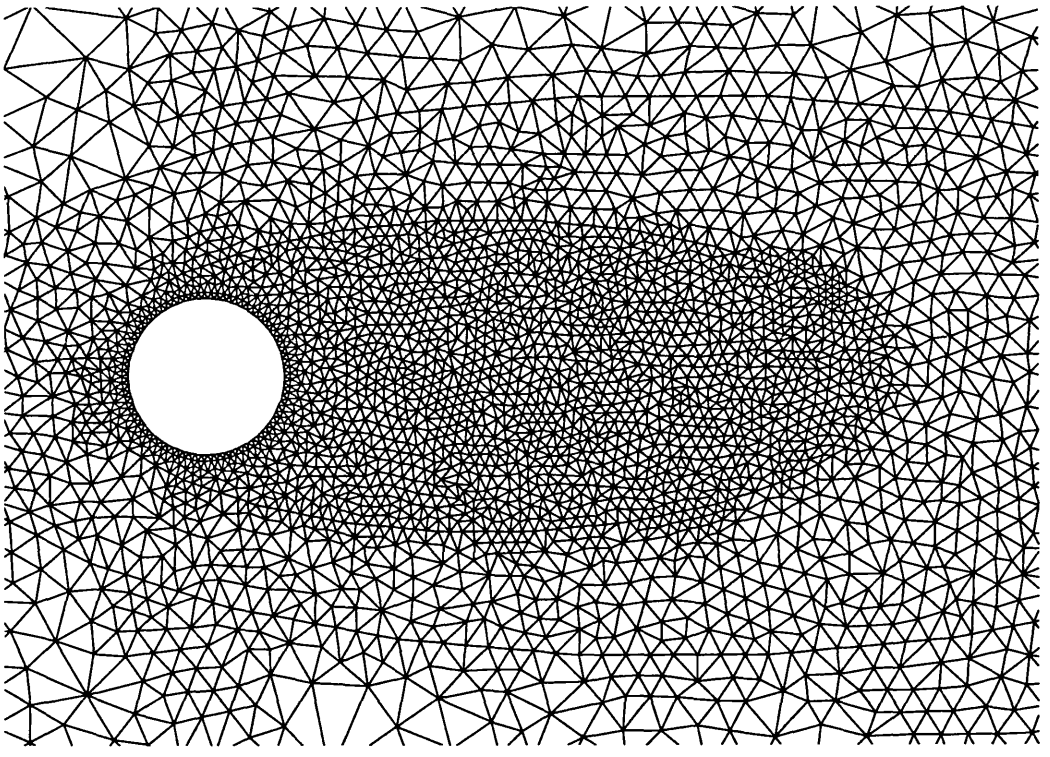


Figure 9.4 Mesh refinement in the wake of the cylinder. This mesh has 80 degrees of freedom round the cylinder. Note only a small portion of the full domain is shown

The discretisation is therefore refined to provide detail where it is desired. Figure 9.4 shows a typical mesh.

Full boundary, grid and time step dependency tests would be desirable although, due to computation time constraints, this is problematic as the Reynolds number continues to be increased. However at Re = 40 analyses were executed for meshes with controlled numbers of degrees of freedom round the cylinder and element sizes in the recirculation zone. These tests, attempting to converge on experimentally observed values for the parameters θ_s and s/d, are given in Table 9.1.

Figure 9.5 was produced at time t^* = 8.8, using the finest mesh described in Table 7.1. The analysis continued beyond this time and the above formation was found to remain stable. The geometry predicted by this simulation thus agrees very well with experimental predictions. Taneda[148] for example gives the values $\theta_s = 53^\circ$ and $s/d = 2.1$.

no. DOF on cylinder boundary	mean separation angle $\theta_s/^\circ$	recirculation length s/d
40	49.0	2.09
80	51.9	2.14
100	52.1	2.22
120	52.2	2.19
Experimental results of Taneda[1]	53	2.1

Table 9.1 Mesh convergence tests for flow past a cylinder $Re = 40$

For very course meshes stability was sometimes not achieved. An asymmetric wake first occurs at $Re \approx 46$ [149]. In this implementation a completely irregular topology mesh is used, and so for courser meshes these asymmetry errors cause premature shedding.

As the Reynolds number is increased beyond this point the velocities in the recirculation bubble increase causing instabilities and asymmetry in the wake. Eventually the Föppl pair are convected, detach and are shed into the wake.

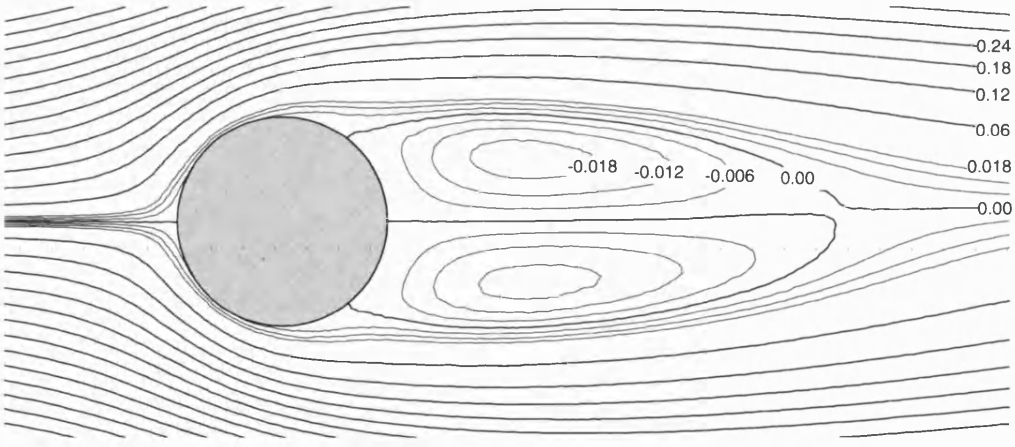


Figure 9.5 Streamlines of flow past circular cylinder $Re = 40$, spacing 0.06 and 0.006 highlighting the Föppl vortices

9.2 Flow past a circular cylinder $Re = 100$

Figure 9.6 is an illustration of the phenomenon of vortex shedding, occurring for flow past a bluff body. As White[8] says this periodic formation is named the von Kármán vortex street following work by Kármán[150] in 1911.

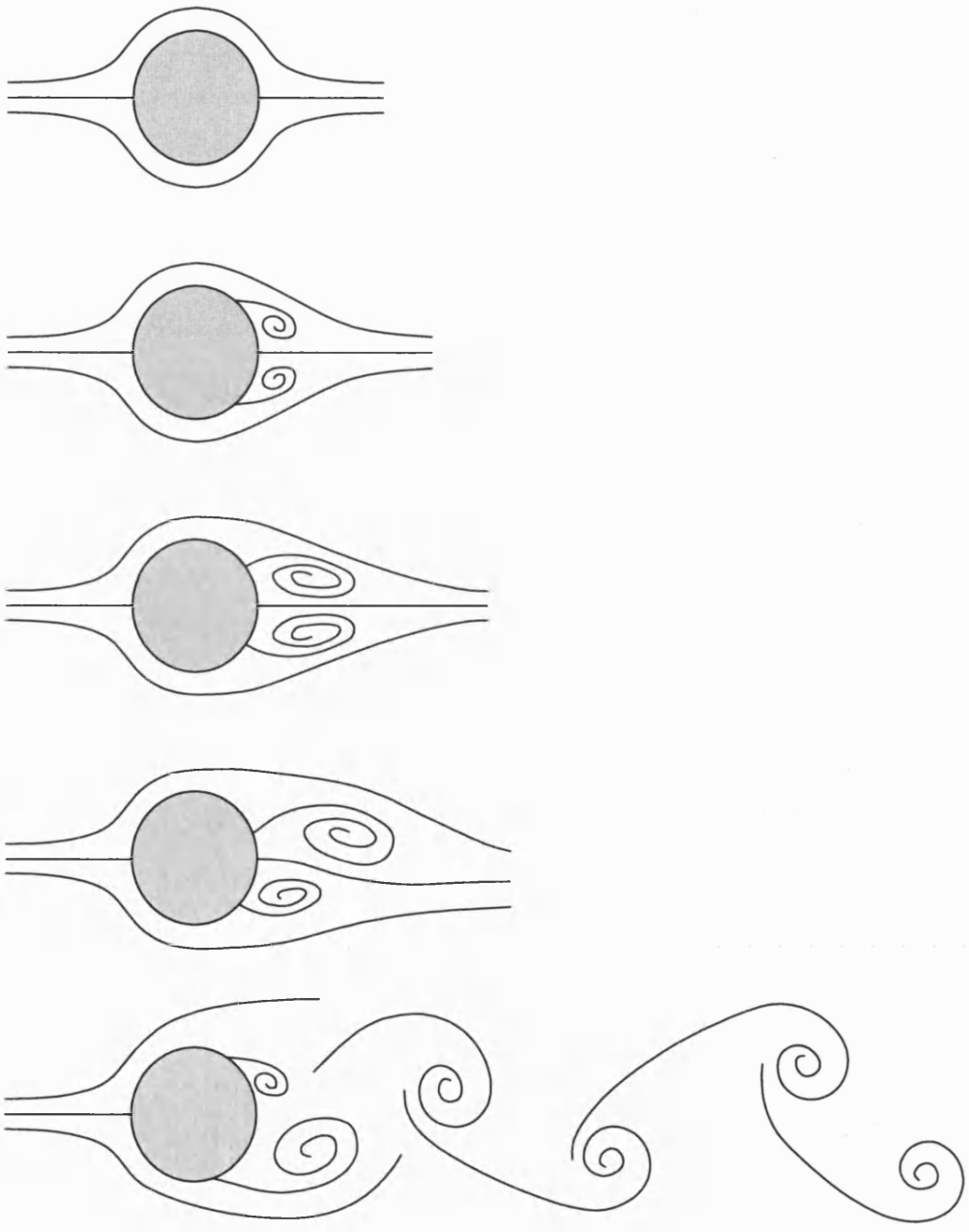


Figure 9.6 Development of the Von Kármán vortex street

To model this behaviour for $Re = 100$, a similar domain geometry to that employed for $Re = 40$ is used. The only difference in the algorithm being the value of μ/ρ and the time step ΔT.

A smaller time step is required in order to ensure stability. The viscosity of the fluid is increasingly dominated by the inertia terms as the Reynolds number increases. The discretisation mesh is increasingly unforgiving because the reduction in viscous damping increases the problem of numerical oscillation, as also observed by Franke *et al.*[151] Figures 9.7 and 9.8 were produced during initial experiments into grid refinement. In Figure 9.7 a well defined refinement zone can be identified around the cylinder boundary and extending into the wake. However the domain elements adjoining this region can be seen to be significantly larger. The convection matrix C is multiplied by a ψ squared term and so small errors quickly escalate. Advection from a number of small elements of the domain to larger elements causes an interpolation to be carried out as potentially degrees of freedom are lost.

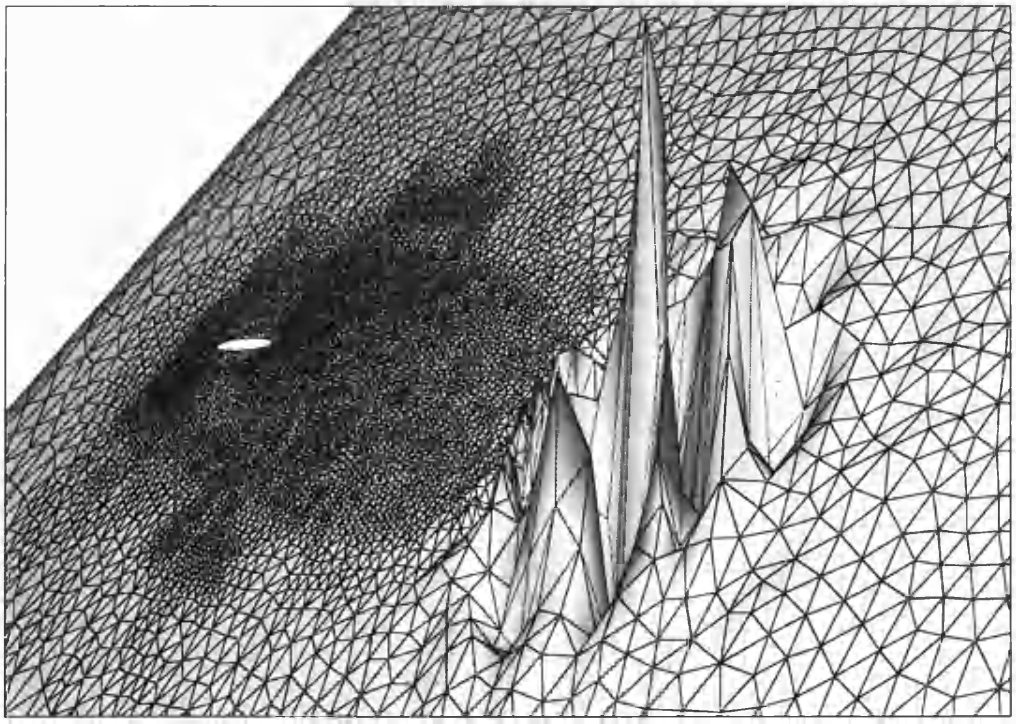


Figure 9.7 Numerical instabilities developing in the domain: Cylinder wake

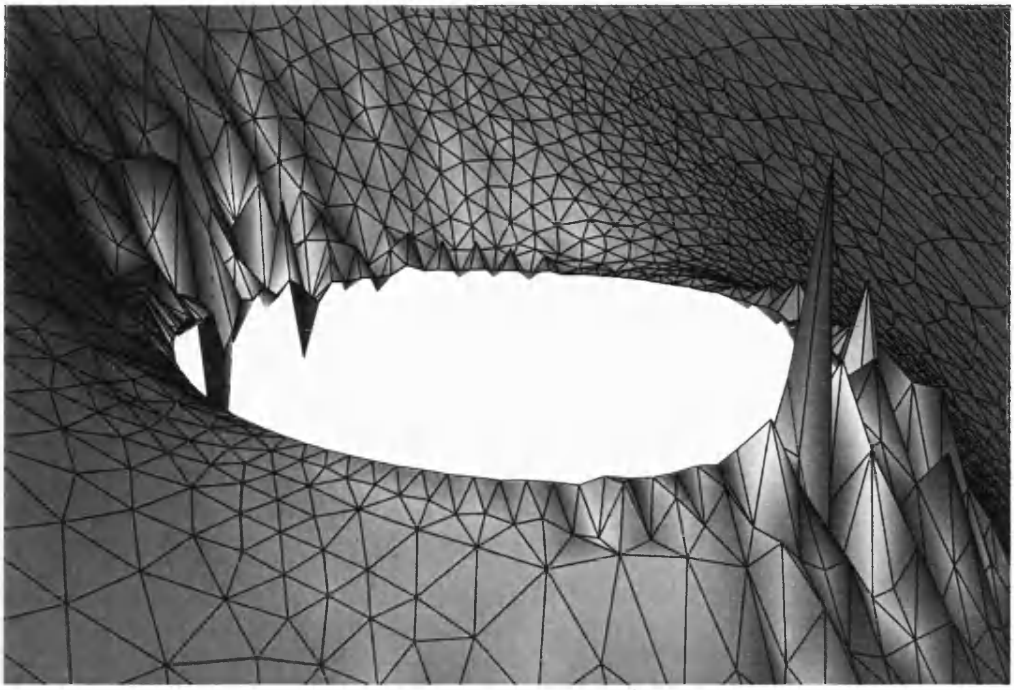


Figure 9.8 Numerical instabilities developing in the domain: Boundary layer

Therefore a sudden transition in element size in a given mesh can cause instabilities in the solution as illustrated in Figure 9.7. Another area vulnerable to instabilities is the boundary layer region shown in Figure 9.8. Here the no-slip condition causes a tight crease to form in the stream function surface which has large curvatures and steep gradients. These sensitivities can be controlled by a smooth transition of element size in the mesh and limiting the value of the time step.

The instability of the von Kármán vortex street occurs in nature as a result of the stochastic, chaotic behaviour of fluids magnifying any minute physical perturbations. Asymmetry occurs in this work due to small errors caused by the irregularity of the mesh. Experiments were performed on meshes constrained to be symmetrical about the horizontal x-axis of the domain. Here the initial symmetrical vortex formation was found to be stable over a much larger range of Reynolds numbers. For symmetrical discretisations Braza *et al.* [152] illustrates the requirement for supplementary perturbations to be introduced in order to initiate vortex shedding.

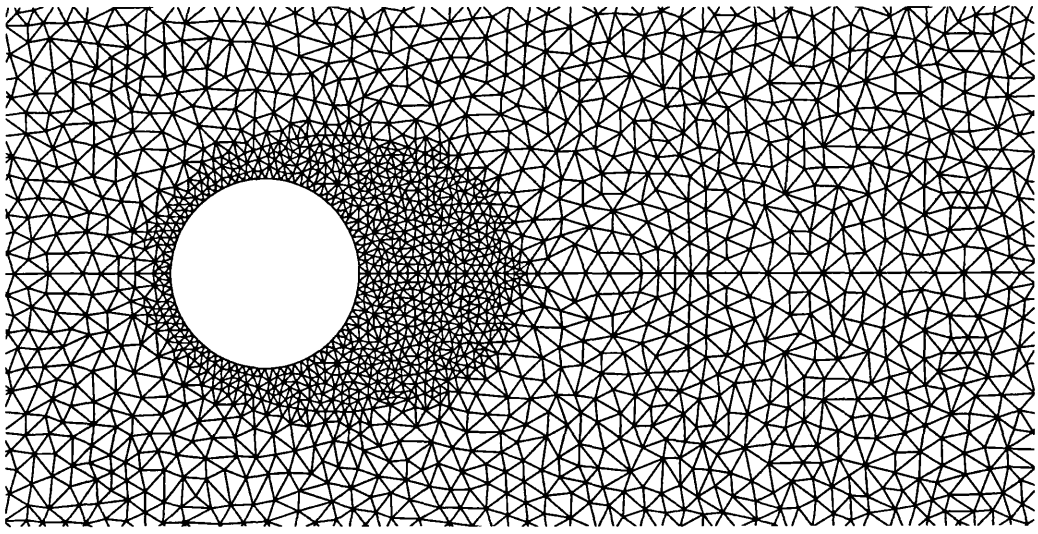


Figure 9.9 Example of an arbitrary mesh, symmetric about the horizontal x-axis. Note only a small portion of the full domain is shown

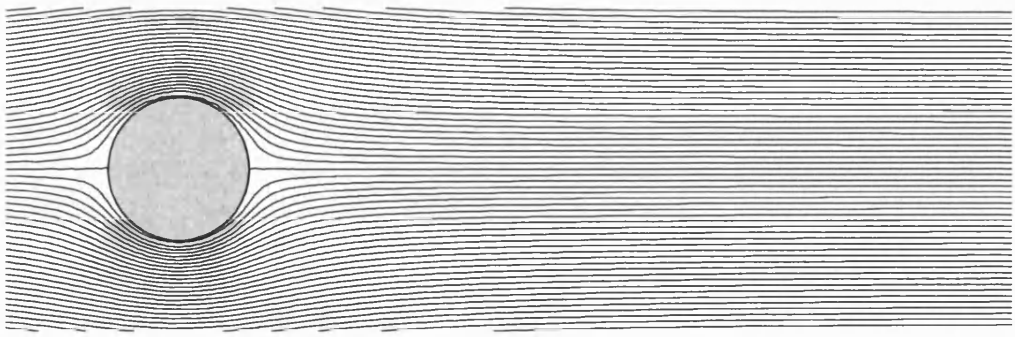
Figure 9.9 gives an example of a mesh employed during these experiments into symmetrical discretisations. However in general fully arbitrary mesh topologies were employed much like Figure 9.4. Therefore no additional artificial perturbations were required to induce the shedding cycle.

Figure 9.10 shows the time history for a $Re = 100$ simulation. Starting from potential flow conditions in Figure 9.10(a), images (b), (c) and (d) show the initially symmetrical Föppl vortices forming and stretching in length.

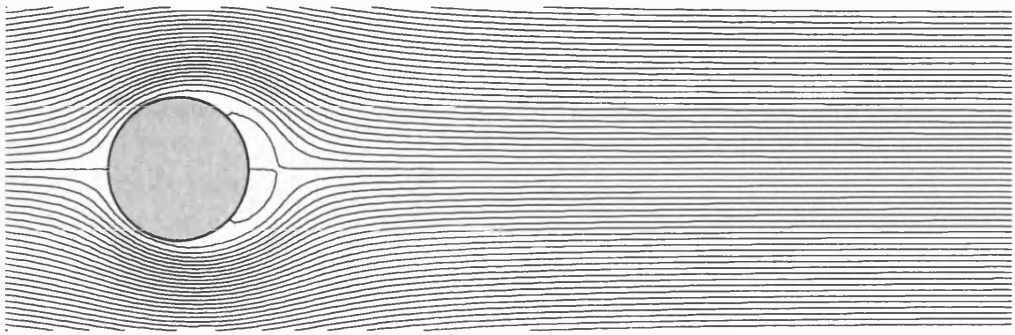
The wake becomes asymmetric after approximately 8-9 nondimensional seconds. Non-dimensional time t^* is given the relationship

$$t^* = \frac{\text{freestream velocity} \times \text{time}}{\text{diameter of cylinder}}.$$

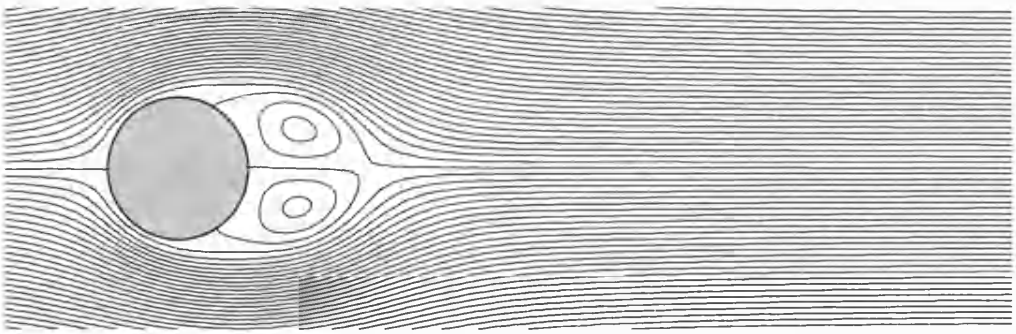
The instability in the wake increases and shedding starts to occur as the Föppl vortices are convected down stream. This can be seen occurring in (e) onwards.



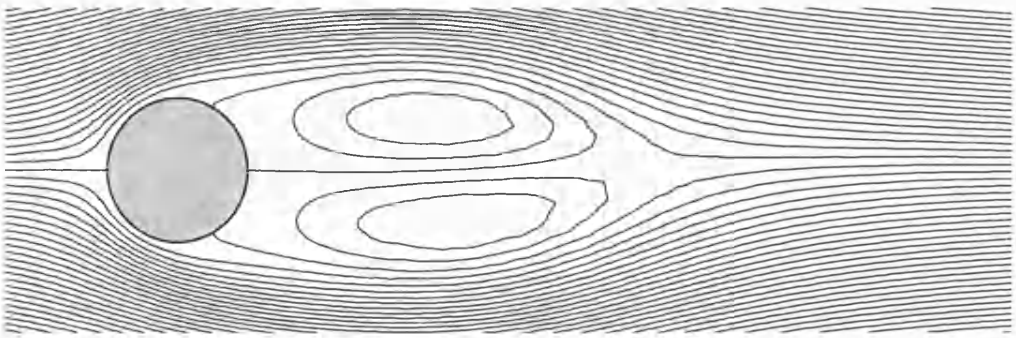
(a) $t^* = 0.0$



(b) $t^* = 0.6$

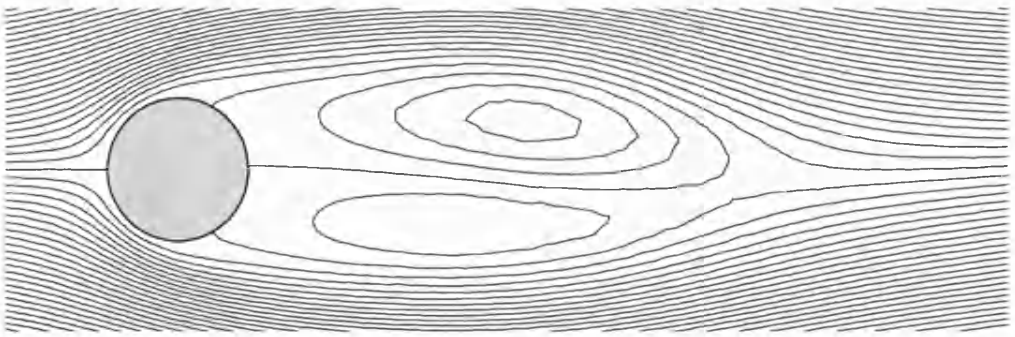


(c) $t^* = 1.8$

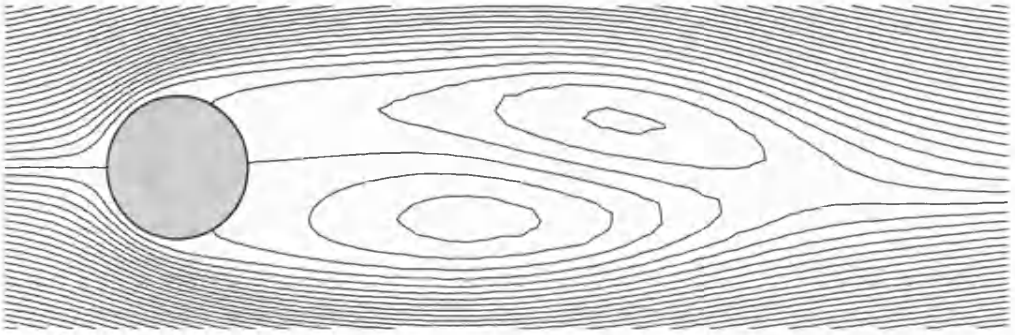


(d) $t^* = 8.0$

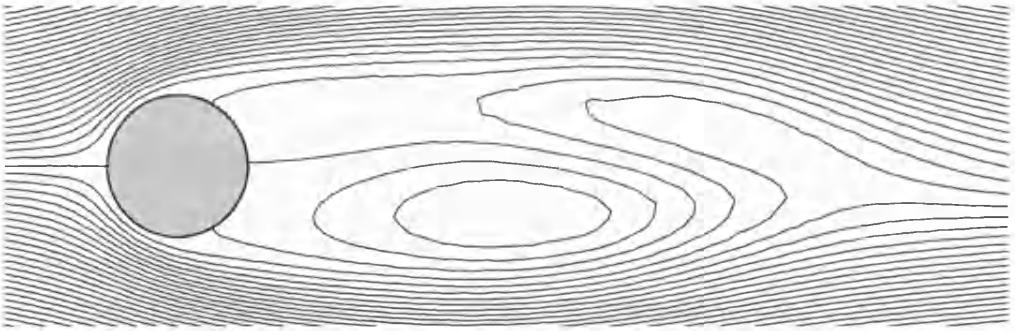
Figure 9.10 Development of the von Kármán vortex street, $Re = 100$, spacing 0.04



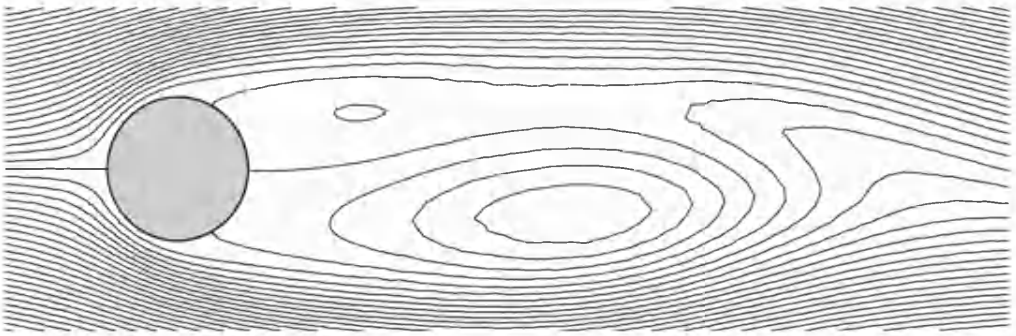
(e) $t^* = 11.1$



(f) $t^* = 13.1$

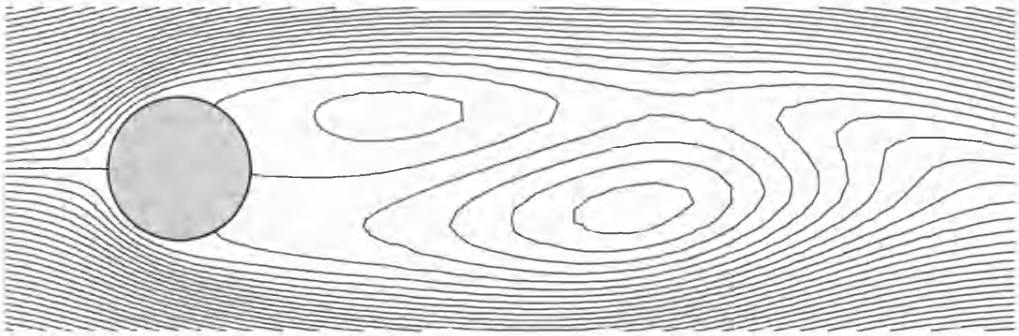


(g) $t^* = 14.1$

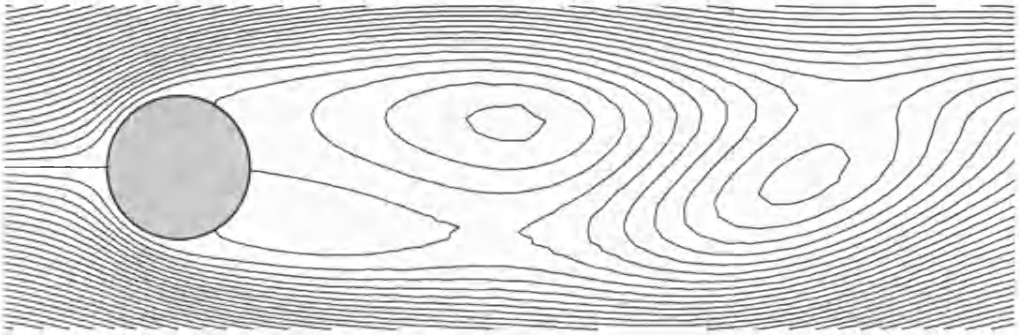


(h) $t^* = 15.1$

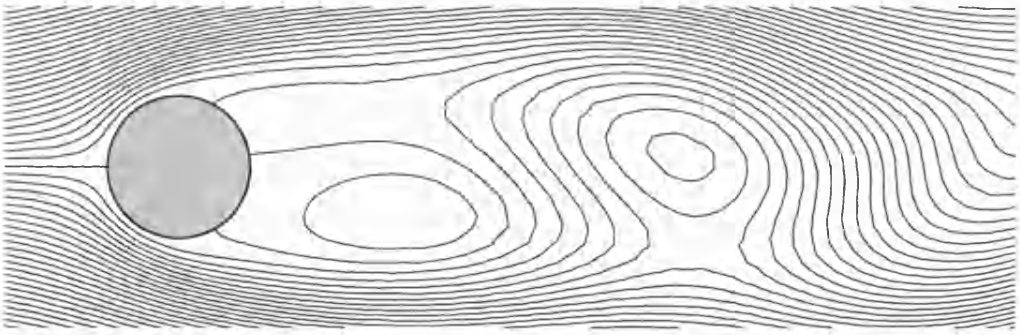
Figure 9.10 continued



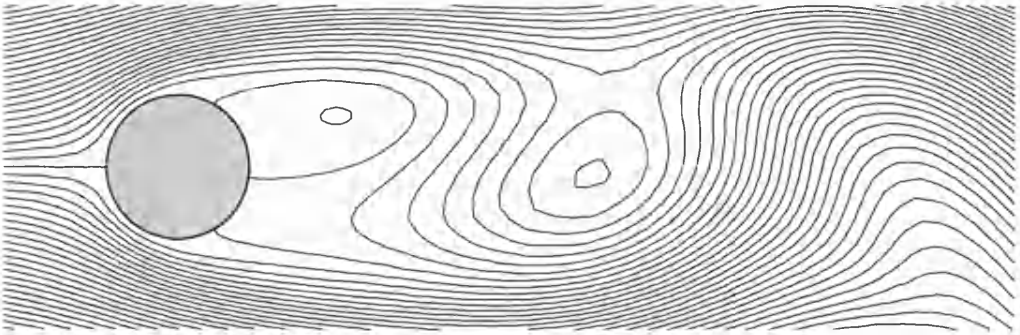
(i) $t^* = 16.0$



(j) $t^* = 18.1$



(k) $t^* = 20.5$

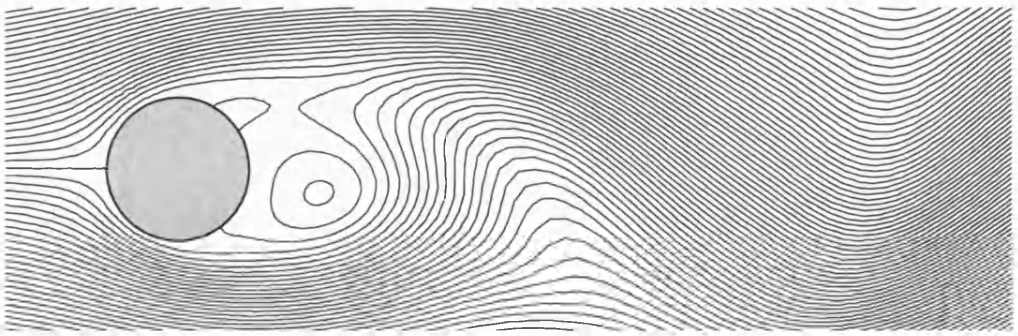


(l) $t^* = 23.3$

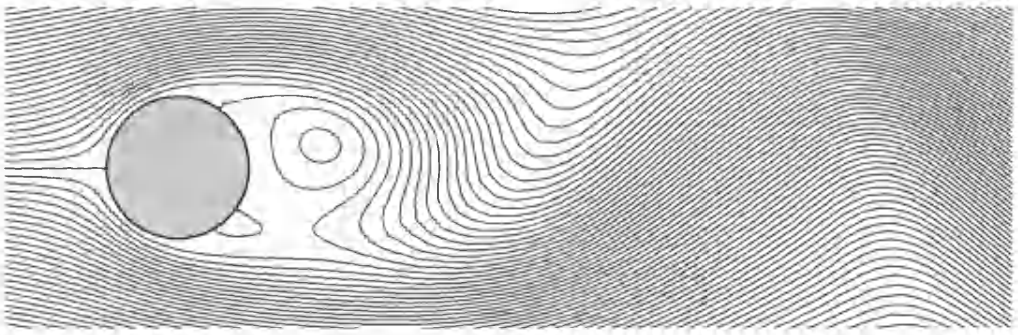
Figure 9.10 continued

After around 20 nondimensional seconds the cyclic process of shedding has begun and after several shedding cycles the oscillatory formation has emerged. Figure 9.11shows one complete shedding cycle.

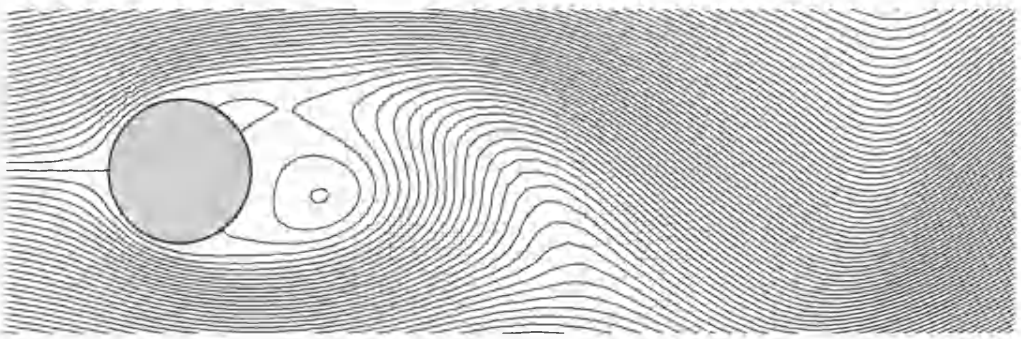
Figure 9.11 and the graph in Figure 9.12 illustrate the vortex shedding occuring periodically, with $T = \Delta t^* = 6.02$, and a frequency (the Strouhal number) of $St = 0.166$.



(a) $t^* = 75.6$



(b) $t^* = 78.6$



(c) $t^* = 81.5$

Figure 9.11 The shedding cycle for fully developed flow $Re = 100$, spacing 0.04

This Strouhal value agrees with existing predictions of shedding behaviour at $Re = 100$. The computed values given by Braza *et al.*[152] and Karanth *et al.*[153] are 0.16 and 0.164 respectively whereas the experimental value of 0.166 is given by Roshko[154].

The point, for which, Figure 9.12 is plotted, is located at approximately $(1.0,-0.15)$ relative to the centre of the cylinder (noting the cylinder diameter is 1.0). Before vortex shedding occurs, between the times $t^* = 0$ and $t^* \approx 15$ the stream function at this point varies in a fairly irregular way.

Looking at Figure 9.10 during this interval one can see this corresponds to the initial stretching of the Föppl vortex pair and their subsequent instability, oscillation and convection downstream. Following a period of increasing applitude of shedding, at approximately $t^* = 75$, the flow is fully developed. Looking at the constant amplitude at the far right of the time axis in Figure 9.12 it can be seen that the implementation has converged on a periodic solution.

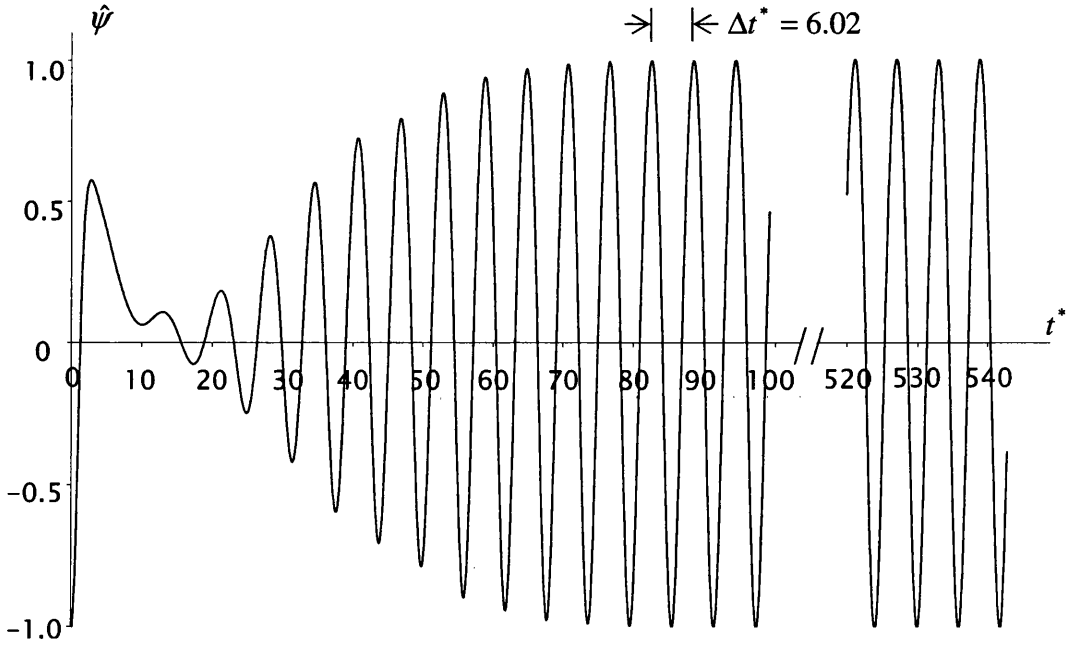
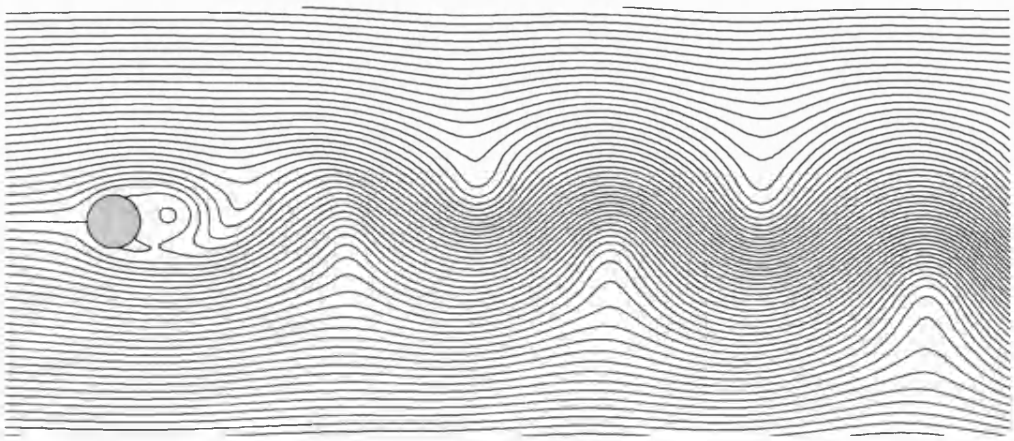


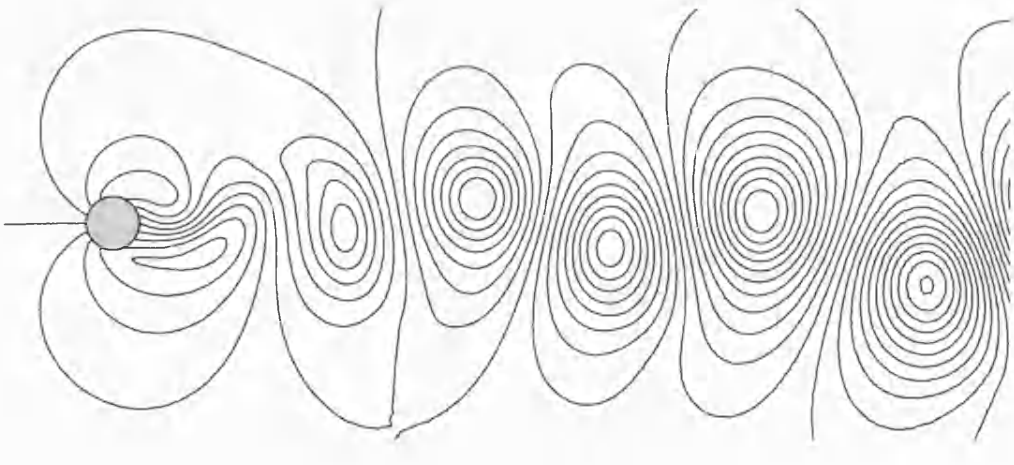
Figure 9.12 Variation of the stream function at a point, normalised with respect to maximum amplitude. *Re* = 100

9.3 Flow past a circular cylinder $Re = 250$

The flow patterns that one experiences are dependant on one's relative velocity to the fluid. The two images in Figure 9.13 are produced from identical simulations, at exactly the same point in time, the difference being the surface contours in the lower image, 9.13(b), have the average velocity of unity removed. In Figure 9.13(a), the observer is stationary with respect to the cylinder. The fluid is therefore seen to be flowing past the body. Figure 9.13(b) is equivalent to an observer watching a moving cylinder passing through initially undisturbed fluid.



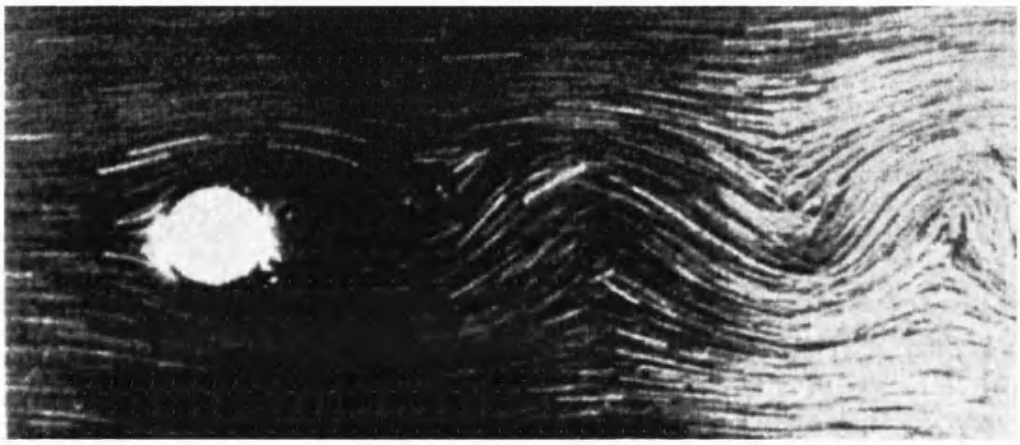
(a)



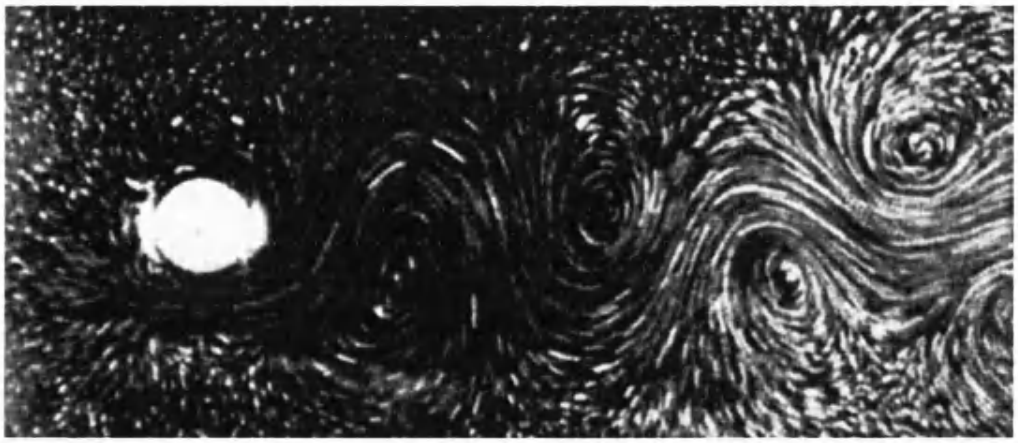
(b)

Figure 9.13 Von Kármán vortex street. $Re = 250$; contour spacing 0.15. Flow past a cylinder (a) moving cylinder in initial stationary fluid (b)

These qualitative flow patterns compare very well with the photographs of Prandtl and Tietjens[147] reproduced in Figure 9.14. The corresponding surfaces are given in Figure 9.15, over the page. Observation with a motion equal to the mean fluid velocity renders the individual vortices comprising the von Kármán street much more clearly. Images for one full vortex shedding sequence are given in Figure 9.16.

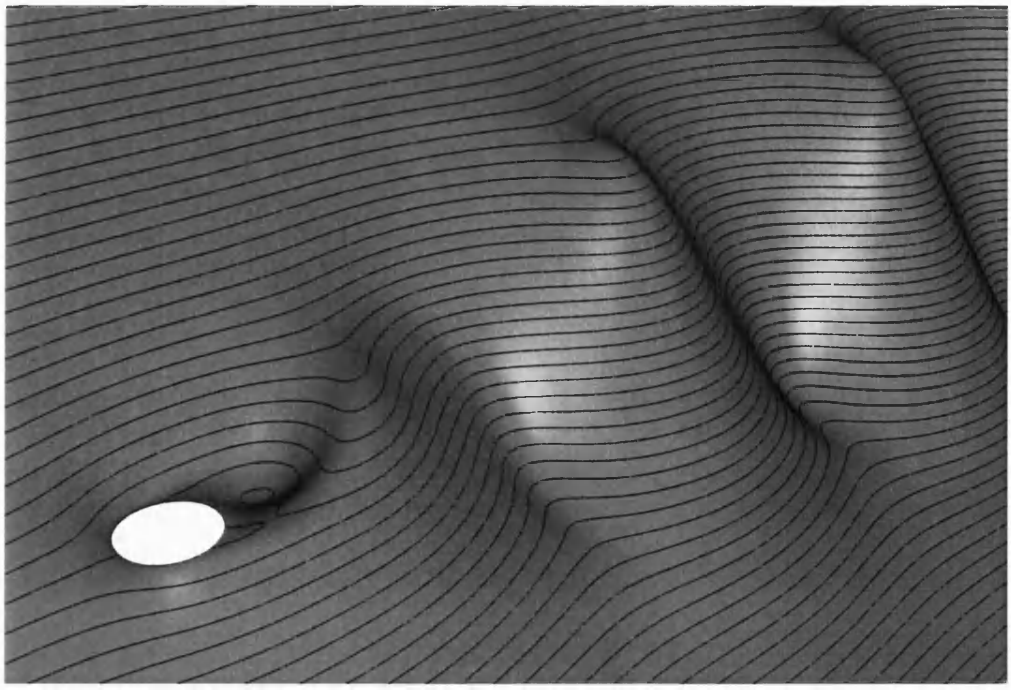


(a)

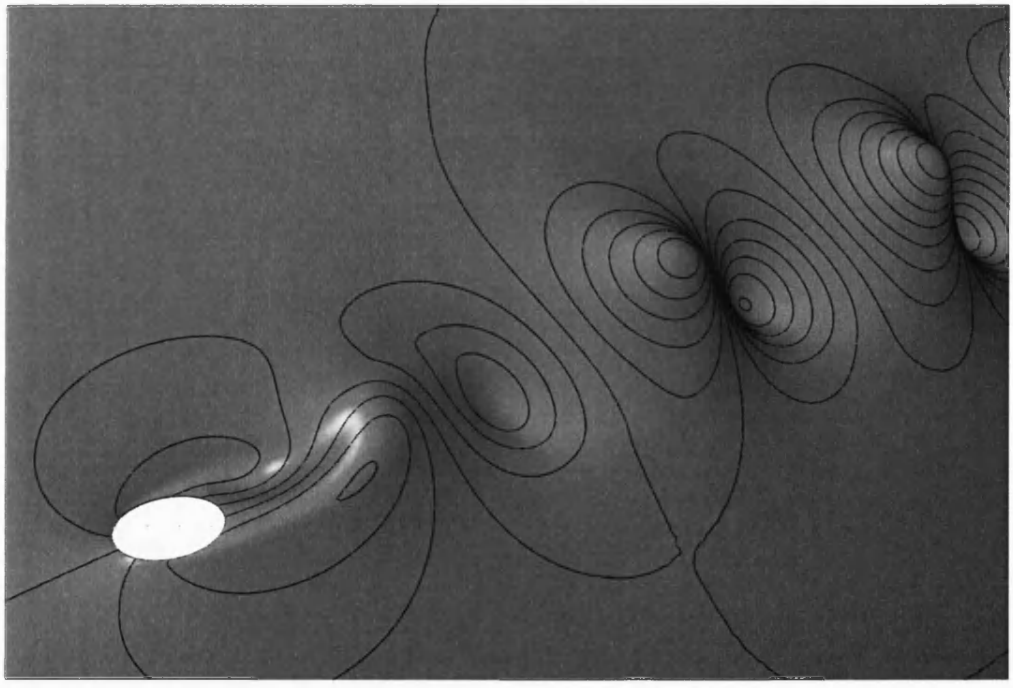


(b)

Figure 9.14 Von Kármán vortex street. $Re = 250$. Camera is stationary relative to the cylinder (a). Camera is stationary relative to the undisturbed fluid (b). Images reproduced from page 303 of Prandtl and Tietjens[147]



(a)



(b)

Figure 9.15 Stream surfaces for $Re = 250$. Flow past a stationary cylinder (a), moving cylinder (b), produced from identical simulations, at identical points in time

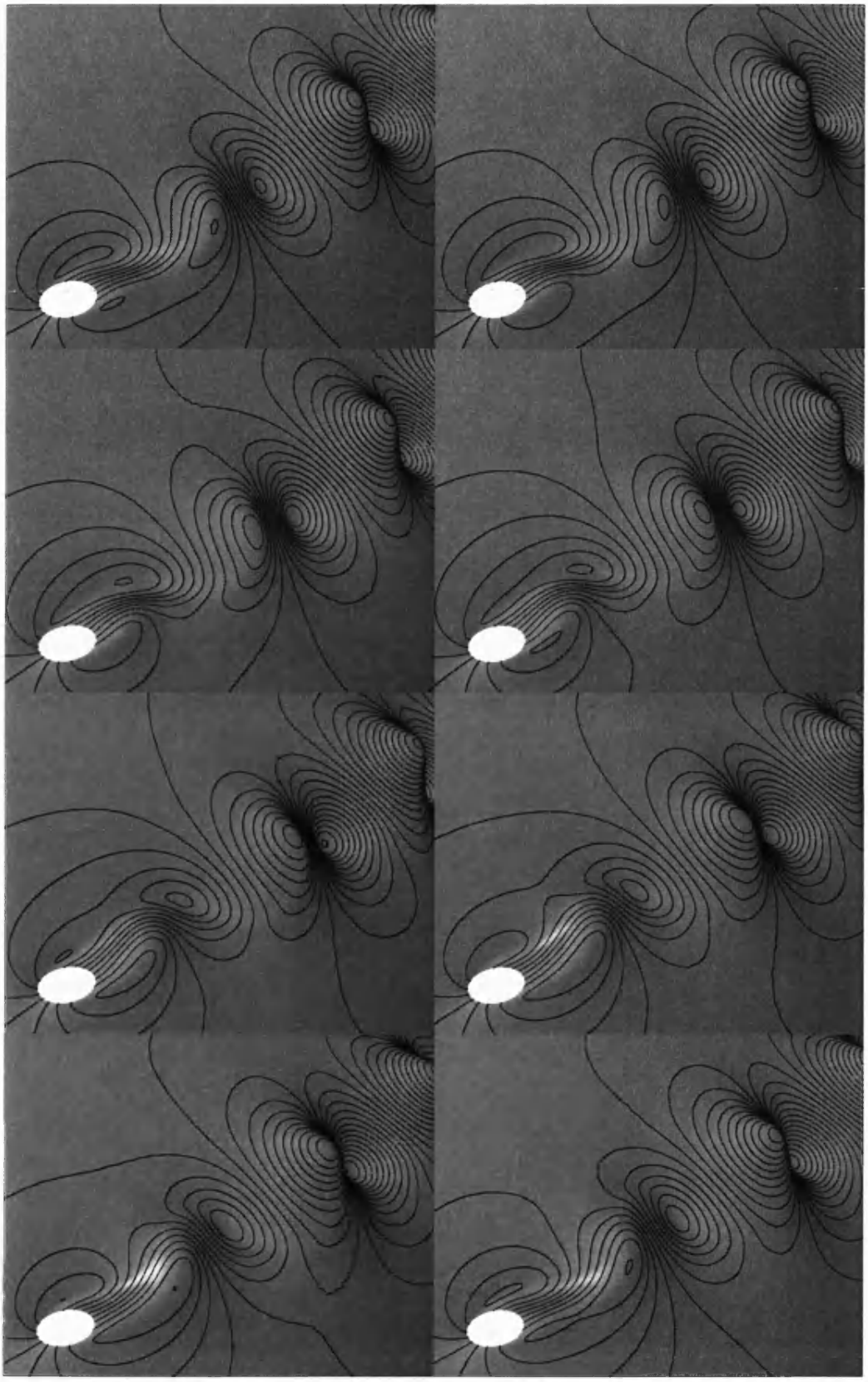
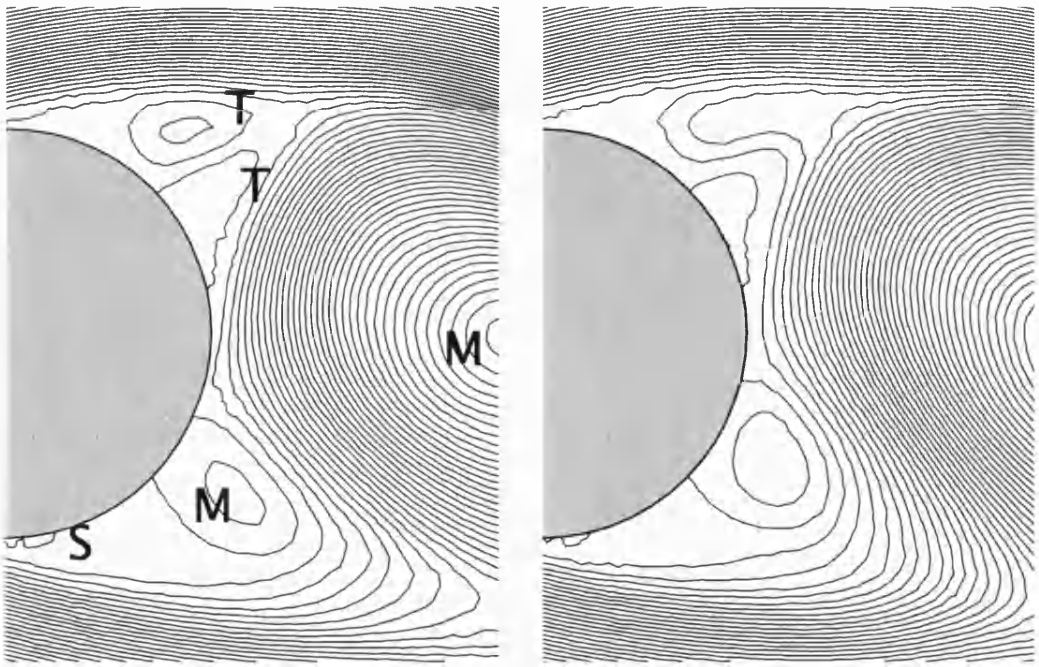


Figure 9.16 Stream surfaces for moving cylinder, $Re = 250$ through one cycle of vortex shedding

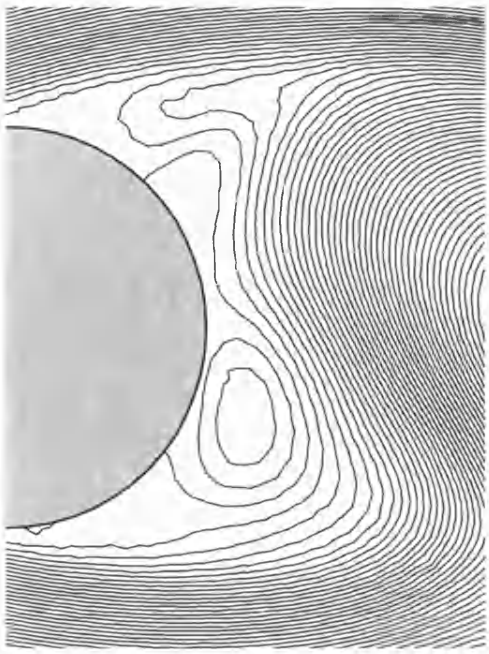
9.4 Flow past a circular cylinder $Re = 1000$

The numerical experiments presented so far have shown good prediction of vortex shedding behaviour. As one continues to raise the Reynolds number increasingly complex behaviour emerges as further bifurcations occur in the Navier-Stokes equations. Braza et al.[155] predict complex vortex motion in the very near wake of the cylinder at $Re = 1000$. Figure 9.17 shows the formation of secondary vortices adjacent to the primary vortices in good agreement with these predictions. Here the three types of vortex given by Braza et al. can be clearly identified. Looking at Figure 9.17(a), the vortices are referred to as denoted by Braza et al. The primary vortices, M, form the von Kármán vortex street. The small vortex, S in the boundary layer, seen at the bottom left corner of Figure 9.17(a), remains, as predicted, attached near the separation point. The third type of vortex, T, can be seen as a secondary vortex pair. This couple, in the subsequent images (b-f), convects and amalgamates with the primary M vortex.

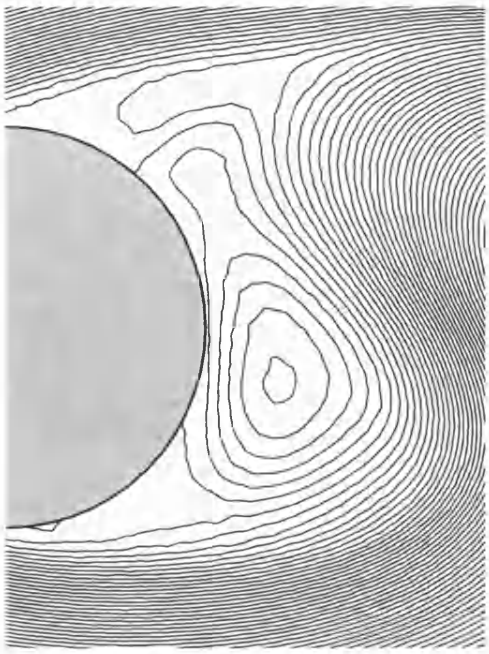


(a) $t^* = 29.2$ (b) $t^* = 29.6$

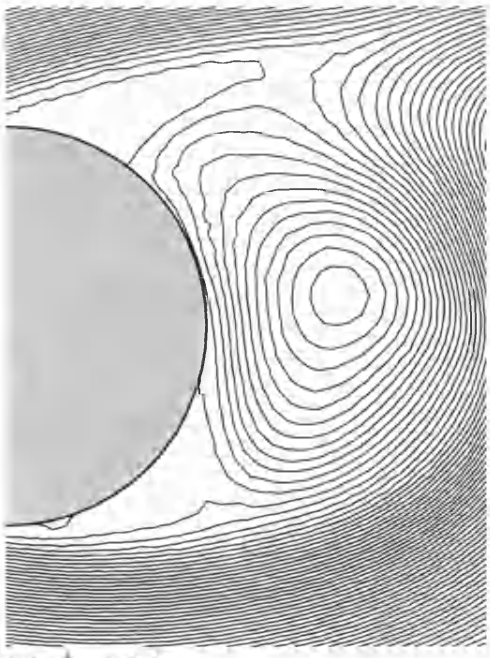
Figure 9.17 Vortex motion in near wake of cylinder. $Re = 1000$; contour spacing 0.014.



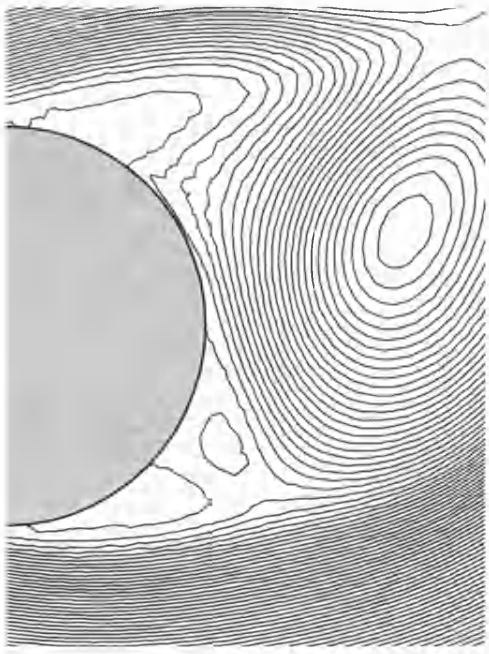
(c) $t^* = 30.0$



(d) $t^* = 30.4$



(e) $t^* = 30.8$



(f) $t^* = 31.4$

Figure 9.17 Continued.

9.5 Flow past a circular cylinder *Re* = 4500

The secondary vortices seen in the previous test case become increasing pronounced and intricate higher order vortices form as the Reynolds number continues to be increased. In Figure 9.18, produced at $Re = 4500$, the beginnings of a hierarchy of vortices can be seen in the boundary layer of the cylinder. Naturally the visualisation at all scales becomes increasingly difficult with streamlines. Experiments with nonlinear spacing of streamlines were carried out, however, although the topology of the streamlines can be expressed more clearly with this method, the contour spacing is not constant and thus valuable information about relative velocities is lost. A useful visualisation technique is used in Figure 9.18 wherein the flow patterns are depicted using velocity vectors. Here the smaller perturbations in the stream function surface are apparent and comparison with photographs of physical results is possible.

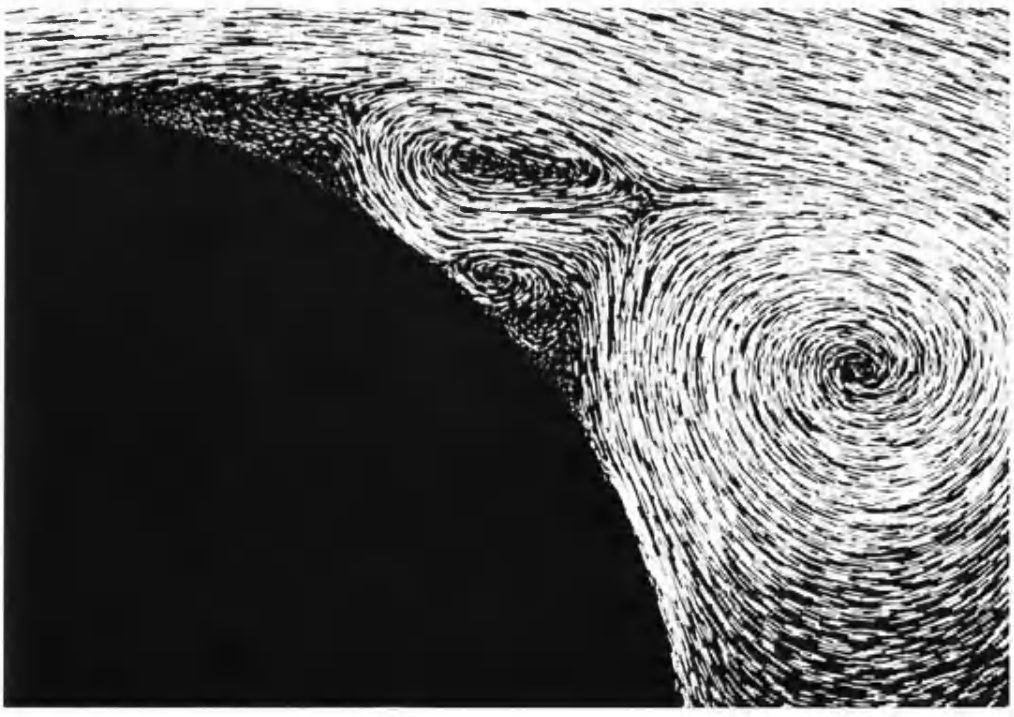
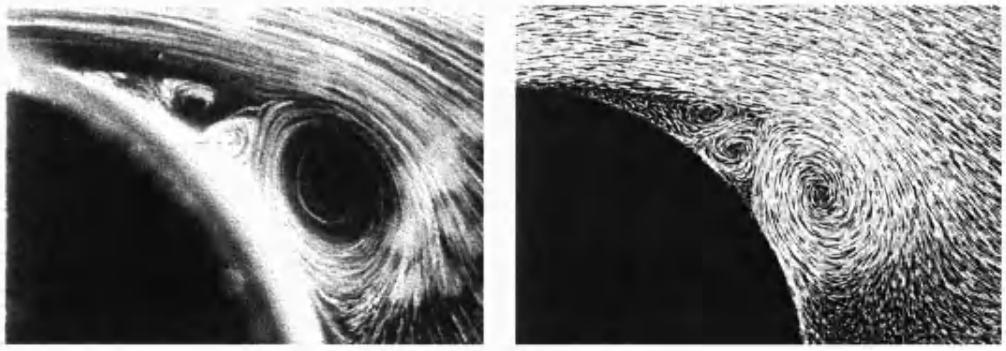


Figure 9.18 Vortex formations during early stages of wake development behind circular cylinder. $Re = 4500$. $t^* = 1.5$. Visualised using velocity vectors generated computationally using the current simulation

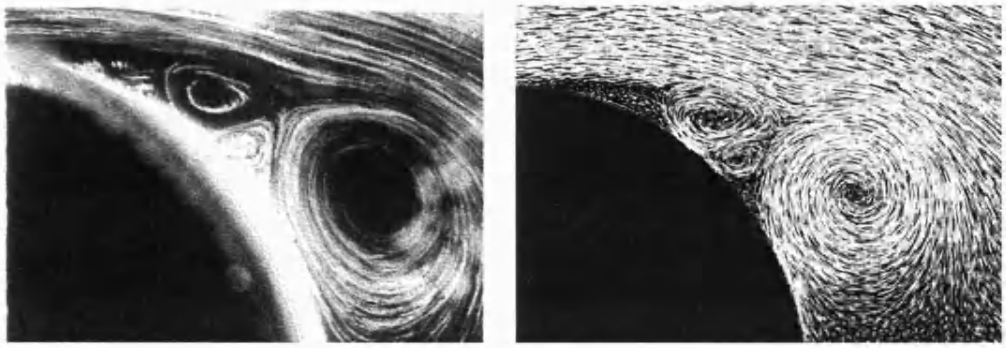
The velocity vectors generated computationally were produced to replicate experimental visualisation methods. Figure 9.19, over the page, shows a series of images capturing the evolution of the vortices produced with the current simulation along with experimental results given by Coutanceau *et al.*[156] and reproduced by Coutanceau and Defaye[157]. The photographs show an impulsively started circular cylinder in an initially undisturbed water tank. Images of this type are produced using a camera travelling such that its relative motion to the cylinder is zero. The flow patterns in the photographs are produced using a short time lapse exposure capturing the short paths of particles of fine powder distributed across the surface of the fluid. A velocity vector field is thus captured, which for short exposure times gives the approximation of the streamline pattern.

The vector images on the right-hand side where thus created to, in part, mimic this visualisation method. Points were randomly distributed throughout the domain and velocity vectors drawn of constant width but with length proportional to the gradient of the stream surface.

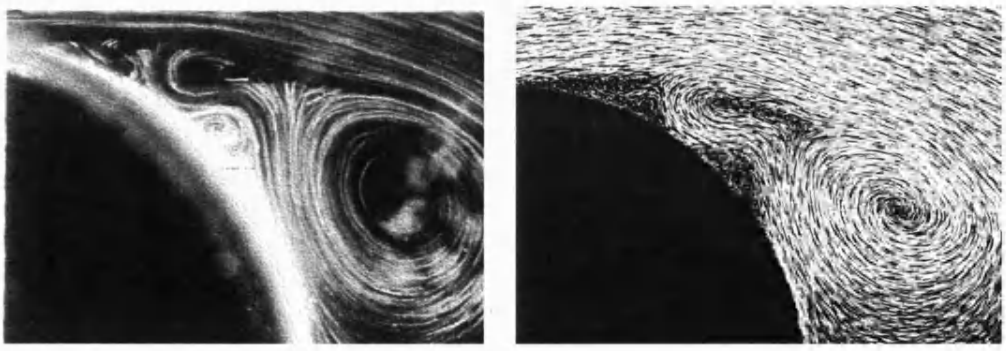
Darker areas in the images should therefore correspond to areas of slow flow, however experimental problems such as the powder particles coagulating in the boundary layer of the cylinder constitute the differences seen between the two simulations in the early stages. Figures 9.19(a-f), neglecting these practical issues, correspond well with the results of Coutanceau *et al.*, predicting the formation of secondary and tertiary vortices and their motion. The convection of the upper secondary vortex of the pair, combining with the primary vortex to form a lobe in 9.19(c) and subsequently merging fully, can be seen to accurately correspond up to image (f). After this point in the simulation, the results given in Figure 9.19(g-h) show that the devolution of the secondary and tertiary vortices as recorded in the physical experiments is not predicted in the computational model. In the current simulation the secondary vortices form a fairly stable vortex pair formation.



(a) $t^* = 1.2$

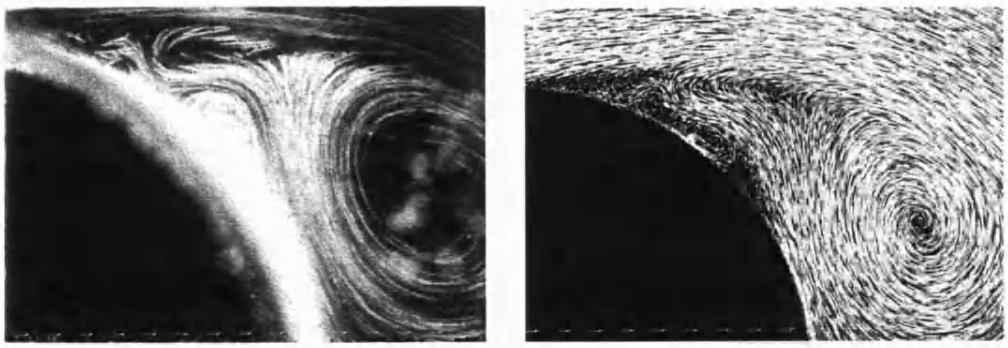


(b) $t^* = 1.4$

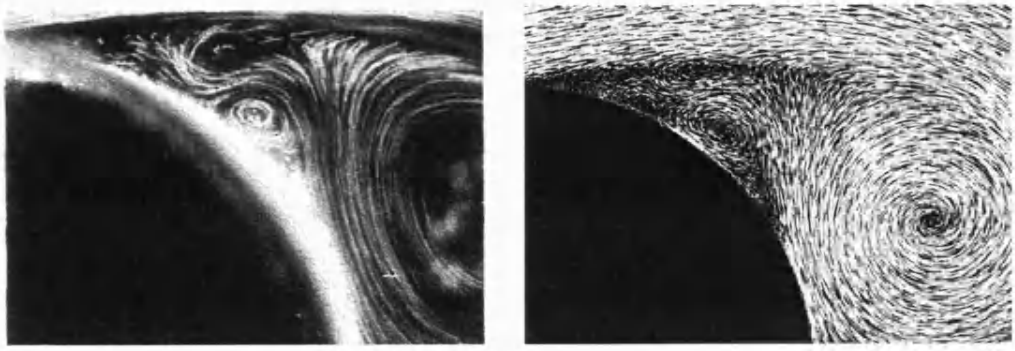


(c) $t^* = 1.6$

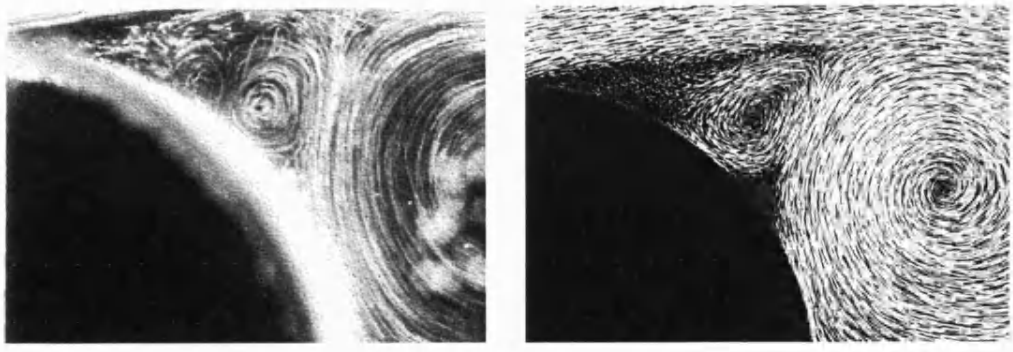
Figure 9.19 Comparison between experimental results (left), reproduced from page 280 of Coutanceau and Defaye[157], and the current numerical simulation (right) for formation of secondary and tertiary vortices. $Re = 4500$.



(d) $t^* = 1.8$

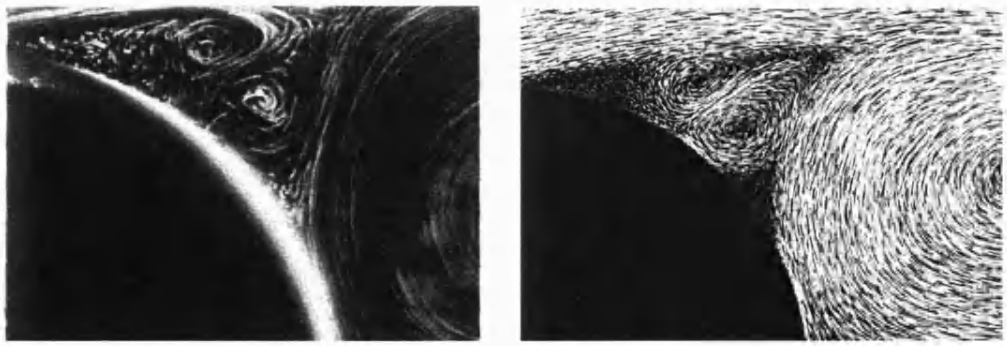


(e) $t^* = 2.0$

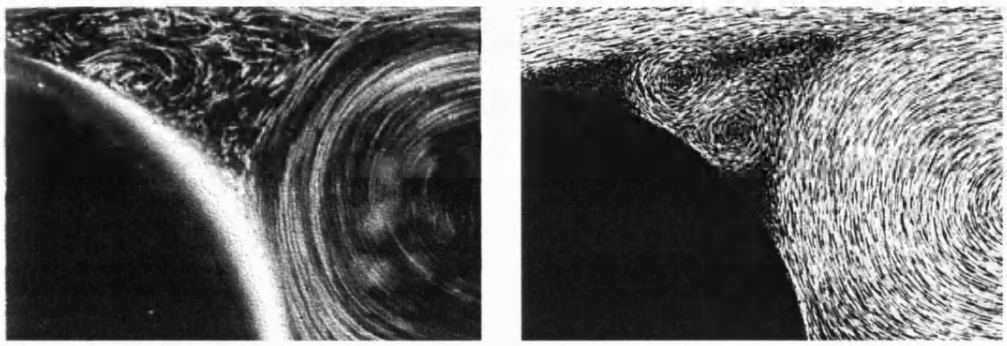


(f) $t^* = 2.4$

Figure 9.19 continued



(g) $t^* = 2.8$



(h) $t^* = 3.4$

Figure 9.19 continued

This decay predicted by the experimental work of Coutanceau *et al.*[156] can be attributed to the three-dimensional effects which, in our two-dimensional simulation, are not reproduced. As Braza *et al.*[155] state, once the Reynolds number exceeds the threshold of approximately $Re = 1000$ the three-dimensional nature of flow can no longer be neglected. In the physical experiments the condition of the transverse velocity $v_z = 0$ is no longer an imposed restriction. These photographs are produced using particles floating on the surface of the fluid.

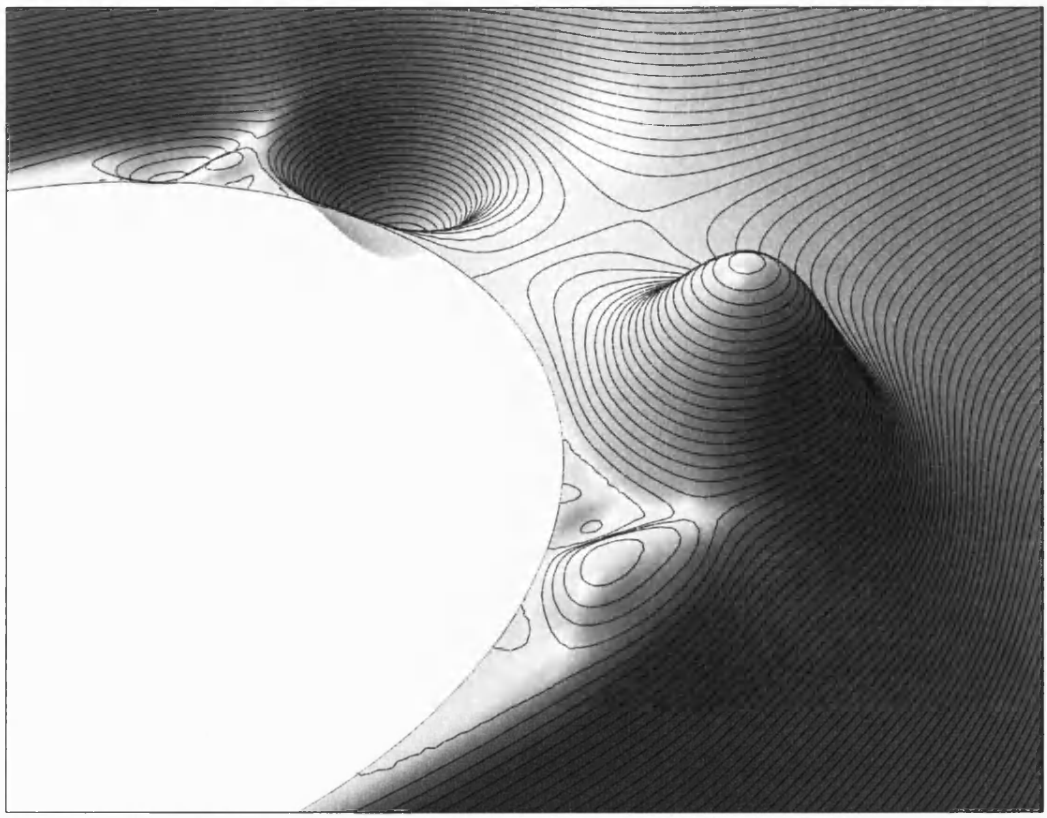


Figure 9.20 Primary, secondary and tertiary vortices in the stream function surface. Flow past a cylinder. $Re = 4500$, $t^* = 1.5$

Figure 9.20 illustrates the gradient and curvature of the stream function surface, highlighting clearly the relative velocity and vorticity in the near wake. Again the sharp fold in the surface forming the boundary layer can be observed. Here, in this image, the self similarity of the vortex formation is evident and the emergence of a fractal-like landscape can be seen.

6.6 Additional test cases

Until now the test cases considered have concentrated on elementary boundary geometries. This has allowed successful verification of the simulations with existing computational and experimental results. However the strength of the derived subdivision finite element methodology lies in its generic applicability to any arbitrary geometry. As stated as objective fourteen, two final test cases are therefore considered to illustrate the methodology's natural ability to handle more complex boundary forms.

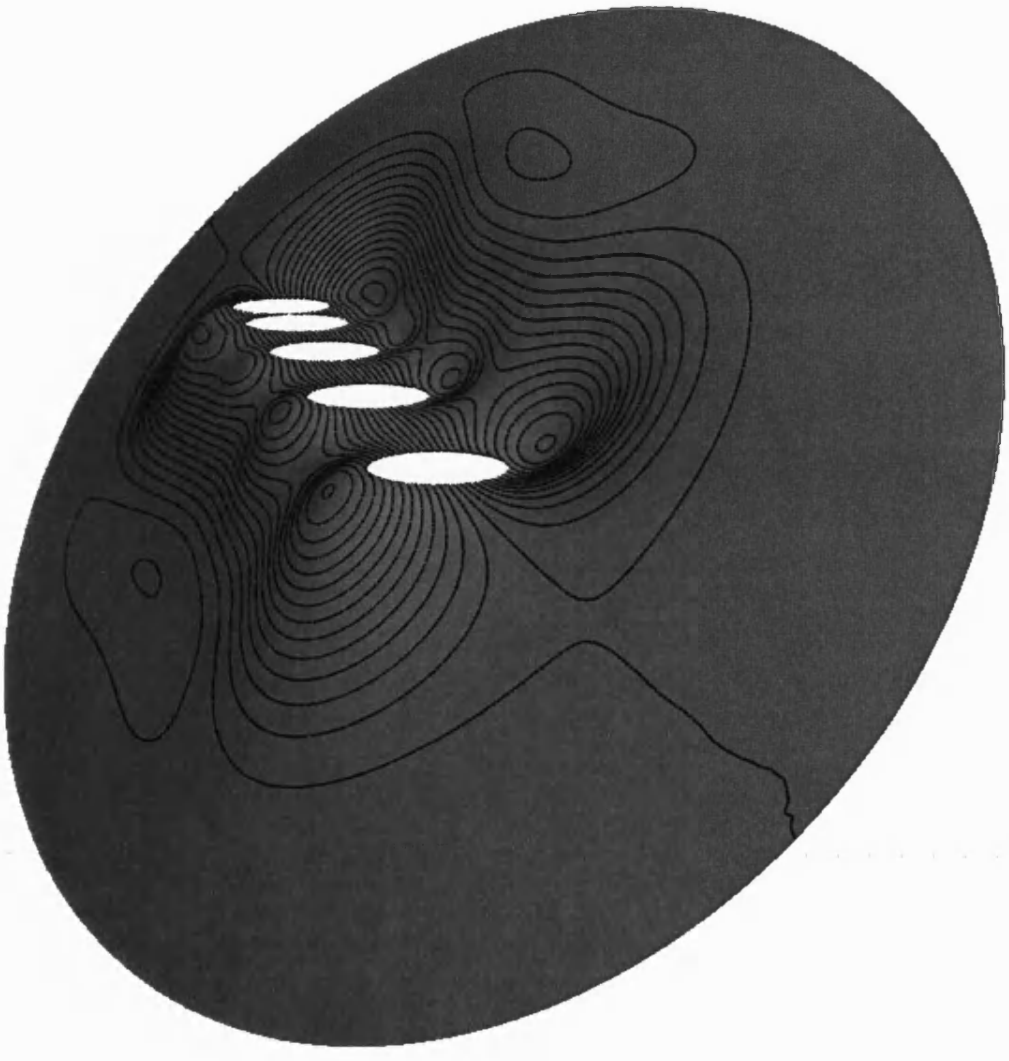


Figure 9.21 The ψ -surface for Stokes flow past five varied spaced circular cylinders. Contours plotted for a stationary observer relative to the fluid

Figure 9.21 shows the implementation applied to the case of five varied spaced circular cylinders. The underlying finite element mesh required to conform to these free-form boundary geometries is therefore highly irregular. In addition, below, in Figure 9.22, the necessary arbitrary mesh topology for a free-form cross-section is illustrated. Here the finite elements are seen to vary greatly in scale through the domain to resolve the detail of the boundary. The resulting surfaces generated during the relaxation of the solution to equilibrium are shown over the page in Figure 9.23.

For both these cases the continuous smooth C^1 conformity of the surfaces is still achieved successfully.

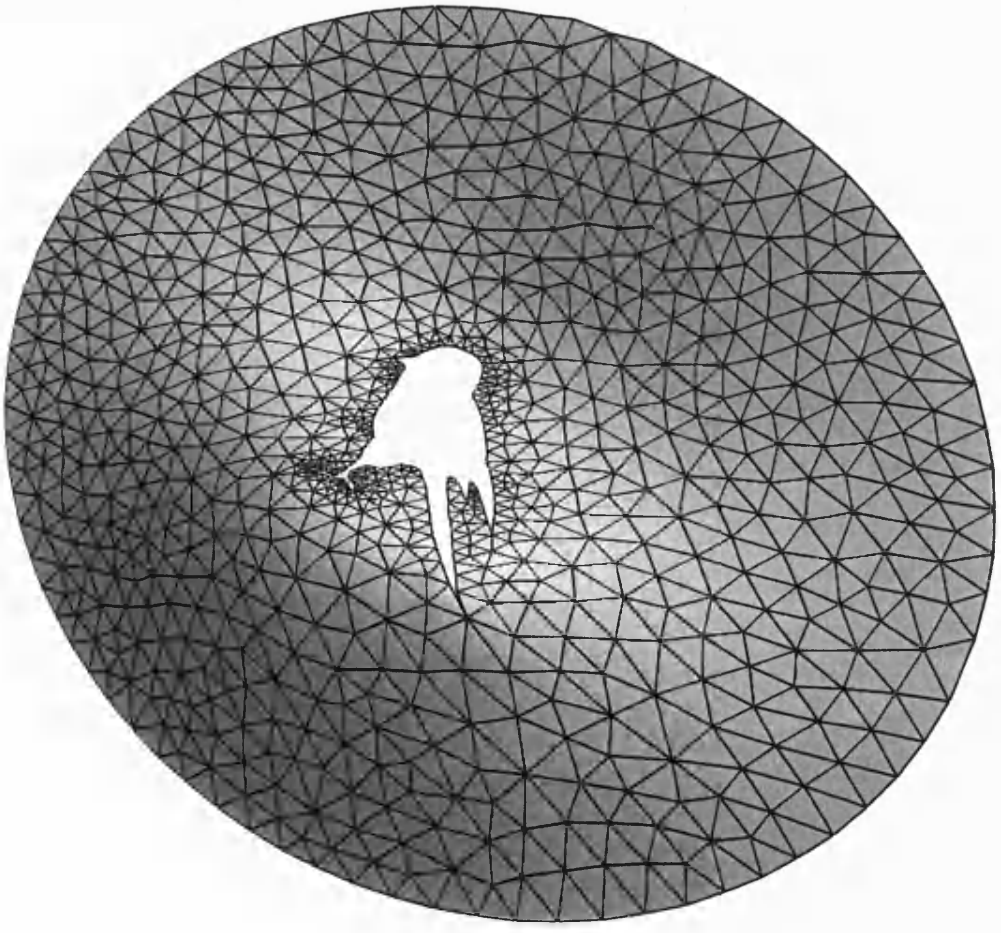


Figure 9.22 Subdivision finite element mesh illustrating high irregularity around arbitrary boundary geometry

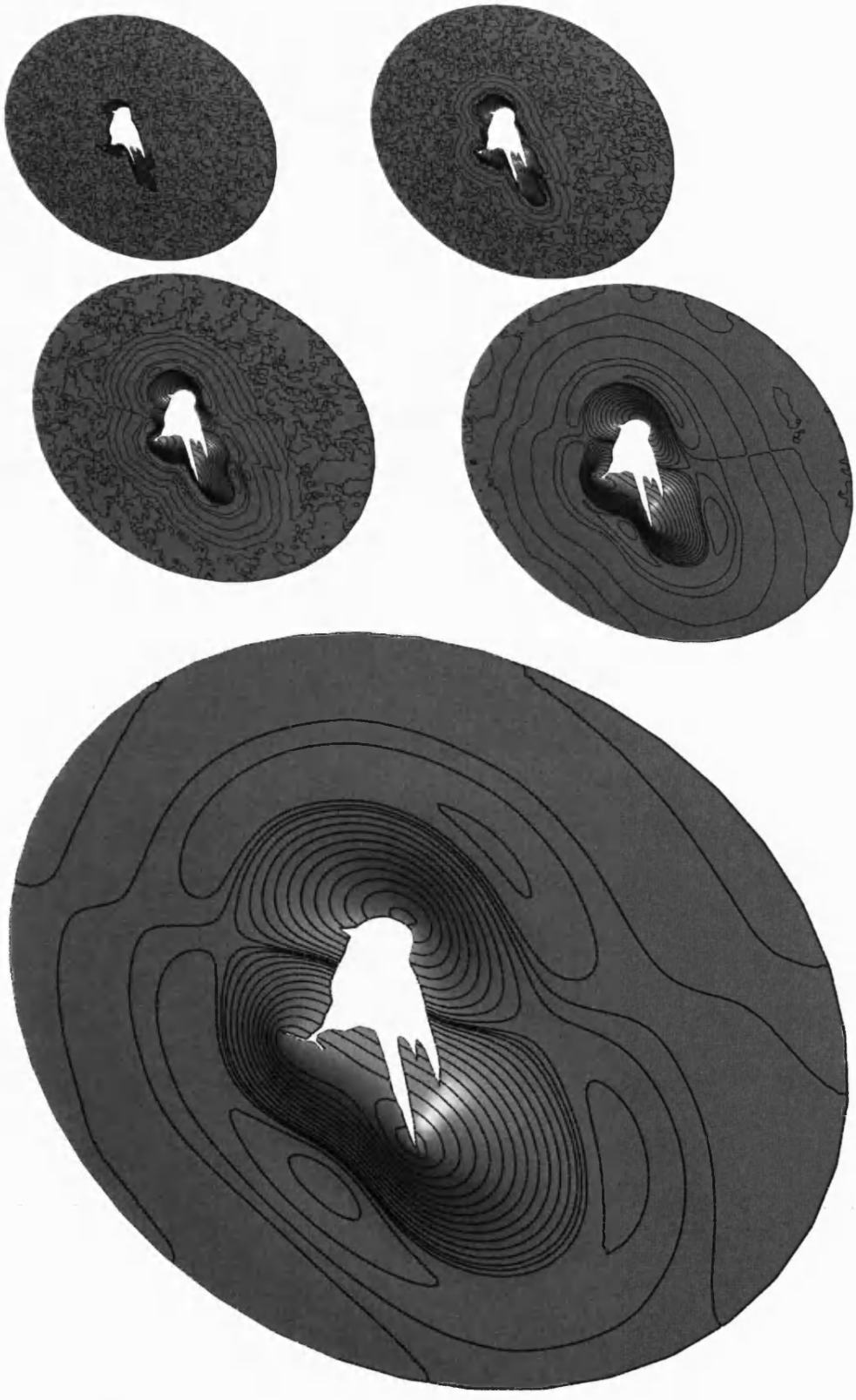


Figure 9.23 ψ -surfaces shown every 25 iterations during relaxation to the Stokes flow solution for flow past an arbitrary cross-section cylinder. Contours plotted for a stationary observer relative to the fluid

9.7 Conclusions

The methodologies developed in this thesis have successfully been applied to a wide range of Reynolds numbers. These results have be verified with very good accordance to existing computational and experimental results completing core objectives eleven and twelve.

As one attempts to model more complex flows at higher Reynolds number, there is increased uncertainty in the accuracy of the results produced. The published data becomes increasingly sparse and grid convergence tests become increasing difficult to perform as run times become measured in days or weeks rather than hours or minutes. Thus within the time frame and scope of this thesis, once the implementation had been validated at low Reynolds numbers, the goal was to reproduce qualitatively the secondary and tertiary vortex geometries generated in the wake of the cylinder. This objective was clearly satisfied by the $Re = 4500$ simulations in section 9.5. Here the intricate vortex formations were produced in very good agreement to existing experimental results.

It was shown that three-dimensional effects can no longer be ignored as one aims to simulate very high Reynolds number flows. There is obviously a limit to which two-dimensional simulations can be employed.

A Reynolds number of 4500 may correspond approximately to drawing a crayon through a puddle†, a little way off modelling, say, wind loads on a building, in which Reynolds numbers in excess of 10^8 can be generated. In order to reach these levels of complexity a turbulence model is required since full direct numerical simulation of all scales cannot be performed.

If one was concerned with a full fluid analysis of the problem a different fluid approach may be more suited. With primitive variables the pressure is

---

† For flow of water μ is 1.0×10^{-3} kgm^{-1}s^{-1} and ρ is 10^3 kgm^{-3}, therefore the Reynolds number $Re = 10^6 vD$ if v is in ms^{-1} and D is in meters.

intrinsically involved in the modelling with the advantageous addition that the methodology can be more readily extended to three-dimensions. Techniques such as large eddy simulation exist in which the self similarly of vortex geometries is exploited such that the smaller higher order vortices can be predicted from knowledge of the larger scale vortices. However in order to build these models a full understanding of the vortices at different scales is needed. This can really only be achieved through a full direct numerical simulation of the behaviour performed at lower Reynolds numbers informing more advanced flows through extrapolation.

It is apparent that the limiting factor on the techniques developed is the computing power available. Direct numerical simulation of fluid behaviour must always suffer from this limitation due the fractal-like nature of the flow geometries, causing one to always strive for more precision and higher resolution. Any discretisation method naturally can only reproduce detail as coarse as the computational grid itself, anything finer is lost. In the case of this thesis the main restriction on processing power was imposed by the convection matrix C which can be seen to be of third order. This is obviously particular to the stream function formulation of the Navier-Stokes equations. Applications of this subdivision approach to other problems would not be met by this restriction.

Finally the potential for this methodology's application to any arbitrary geometry was illustrated through several additional test cases.

Chapter ten

Conclusions and recommendations

10.0 Introduction

The aim of this thesis as written in section 1.1.1 was to formulate an approach for solving highly non-linear high order differential equations in two dimensions thus generating interesting and complex surface geometries. This was achieved through the solution of the well known Navier-Stokes equations expressed in a stream function formulation.

A novel approach to discretising this fourth-order equation using an arbitrary topology mesh has been successfully derived. The high level C^1 continuity of the solution demanded by the viscous or bending stiffness terms was achieved using a new extension to existing subdivision methodologies. This approach was implemented over a range of test cases.

The goal of simulating complex surface geometries was clearly met as intricate vortex formations and motions were observed to naturally emerge from the governing equations.

10.1 Conclusions

The central aim of this research has been successfully accomplished. This has been achieved through comprehensive realisation of the thesis' core objectives identified in the chapter one. The objectives were fulfilled as follows:

i. In chapter three the necessary differential geometry relations were derived successfully. A curvilinear tensor notation is used enabling efficient description of the equations in an invariant form useful for the derivation of the curvilinear subdivision finite element. These differential geometry results are then utilised throughout this thesis in the derivation and the solution of the fourth-order differential equations.

ii. The two partial differential equations of interest in this thesis, the biharmonic equation of plate bending and the fourth-order stream function equation have been successfully derived in a generic curvilinear tensor form.

iii. The analogy between viscous flow of fluids and plate bending was then identified though comparison of the two governing equations.

iv. The existing approach for defining smooth surfaces on an arbitrary triangular mesh proposed by Loop[38] was introduced. This subdivision methodology gives the required C^1 continuity necessary to enable interrogation of second differential information of the limit surface.

v. A novel generic subdivision matrix was devised in order to relate the limit properties of the continuum defined by twelve regular control points to the arbitrary number of global degrees of freedom forming the control

mesh. The subdivision matrix topologically refines the arbitrary finite element to create a local patch of regular mesh which can then be interrogated using twelve regular shape functions.

vi. The weighted residual form of the stream function was then written. This discretised equation was then expressed in an original form using the proposed subdivision shape function applicable to arbitrary meshes.

vii. Dynamic relaxation was introduced as a solution procedure. This methodology was applied to solve the differential equations avoiding the formation and multiplication of the full matrices. Previously subdivision shape functions have been successfully applied to structures[84, 89, 92] in which the methodology has been to form the stiffness matrix and perform an inversion, solving directly. In this thesis the author has successfully shown that relaxation methods can be applied proving this approach to be efficient by removing the requirement to formulate whole matrices. Here these fine finite element meshes would give sparse largely zero matrices, the multiplication of which has been avoided.

viii. Validation of the newly devised arbitrary subdivision element methodology was successfully achieved using three linear biharmonic test cases. The finite element was seen to perform well, converging on the solution efficiently under successive refinements of the meshes.

ix. Approaches to the application of boundary conditions were successfully set out and their viability tested. Boundary conditions are critical in computational fluid mechanics. The outer boundaries of the domain were found, as expected, to have an adverse effect on the results such that the overall domain size needed to be large to avoid boundary effects. The proposed relaxation methods demanded novel rules for imposing the boundary conditions to be developed. These proved successful. Different approaches to enforcing the no-slip condition through ghost nodes where tested. Here the author imposed relationships between the ghost nodes and the internal mirrored nodes, defining the conditions at the

boundary. A novel approach for the application of the cylinder boundary conditions has been devised. An approach for defining the initial conditions for the cylinder and fluid domain were also defined. When implemented this gave good accordance with existing results.

x. An approach for the solution of the non-linear stream function equation was devised. This new methodology consists of a temporal discretisation, each iteration of which forms an iterative relaxation approach to form find the geometry of the Poisson's equation at each given time step. For temporally nonlinear cases such as those used here, after an initial period of egression from the starting configuration, only small changes in geometry are experienced between each iteration, and thus relaxation from the previous state to the next can be achieved rapidly. With direct solution methods a full matrix multiplication is required at each time step. The relaxation method thus proves to be efficient.

xi. The implementation devised in this thesis was applied to the test case of flow past a circular cylinder for Reynolds numbers of Re = 40, 100, 250, 1000 and 4500. The images in chapter nine clearly show the success of the implementation in generating complex surface geometries.

xii. The simulations at the various Reynolds numbers were successfully validated against existing results. Comparison has shown the methodology to be extremely accurate in simulating this behaviour in good accordance with computational and experimental findings.

xiii. A primary goal of this research, set out in core objective thirteen, was to reproduce qualitatively the secondary and tertiary vortex geometries generated in the wake of the cylinder. Comparison was made to photographs produced during physical experiments at Re = 4500. The self similar vortex formations were successful reproduced computationally forming a clear correlation to the experimental predictions. The subsequent devolution of the vortices seen in the physical experiments was not predicted by the simulation, showing that

three-dimensional effects can no longer be ignored to simulate very high Reynolds number flows. There is therefore a limit to which two-dimensional direct numerical fluid simulations can be employed.

xiv. Several test cases were performed on arbitrary geometries to highlight the generic application of this approach to any complex form. The implementation proved to successfully produce intricate smooth surface geometries of constant tangency on the necessary highly irregular meshes.

This thesis has successfully demonstrated the effectiveness of the proposed subdivision finite element method for modelling complex geometry surfaces involving changes in resolution over arbitrary meshes. The problem of fluid flow was chosen specifically because it demanded this requirement. The challenge was to use subdivision to pick up the small perturbations in the stream function describing vortex motion. This has been realised modelling fractal-like vortices forming in the boundary layer of fluid flowing past a cylinder.

10.2 Further work

Potential areas for further work extending this methodology are now presented.

The generation of the underlying meshing is an area in which additional development would be beneficial. The method proposed can handle a completely arbitrary mesh and so a simple constrained Delaunay triangulation was conveniently used with very primitive restrictions on the mesh. A further improvement might be to try to restrict the irregular nodes as far as possible, creating large expanses of regular mesh wherever possible.

Dynamic adaptivity of the mesh during simulations would also be an effective expansion of the existing code. Refining the mesh topology based on the vorticity of the surface, in other words introducing localised subdivision in areas

of high curvature where increased elements are desirable to reveal the small perturbations would be useful. Amresh *et al.*[108] present possible work in this field. Loop subdivision is well suited to handling such approaches without the problem of hanging nodes encountered with regular meshes as illustrated in Figure 10.1.

Conversely decimation[109, 110] could be used in near planar regions of constant velocity. Implementations such as this would require further grid convergence tests on the effect of irregularity in the mesh. Instabilities are generated in convecting between fine areas to coarser areas of the domain, as shown in chapter 9, Figure 9.7.

An area which needs to be explored is the effect of different quadrature methods on the convergence of the solution. This implementation has successfully shown one-point quadrature to be sufficient in agreement with previous studies[84]. However in parallel to exploring different meshes, more complex quadrature methods may enable fewer elements to be required. For spatially linear problems such as those studied here, the matrix evaluation takes place once at initialisation, thus performance would not be adversely affected by any more elaborate integration methods.

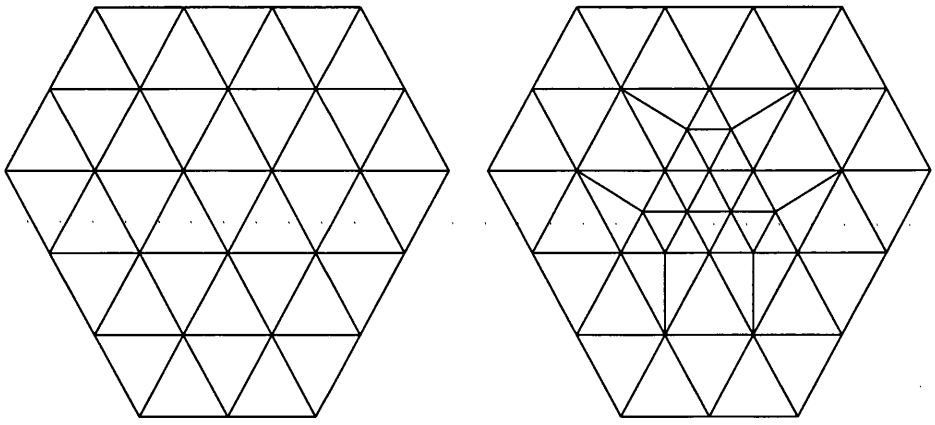


Figure 10.1 Illustration of localised subdivision generating irregular topology mesh whilst negating the requirement for hanging nodes

As one continues to try and model higher and higher Reynolds numbers, efficient use of available computing power is paramount. Experiments into exploiting geometric or algorithmic parallelism[158, 159] would be beneficial as the number of finite elements increases.

Other properties of the surface can be interpolated using subdivision. The example given by Piponi and Borshukov[160] is to use subdivision to smooth texture mapping information over the surface. A five-dimensional coordinate is used, containing two texture coordinates in addition to the three-spatial coordinates. In the case of the stream function surface additional properties such as pressure or vorticity could be calculated and smoothed.

An obvious extension to this work would be to accommodate the third dimension. This is not straightforward with regard to either the governing differential equation we have been solving or the underlying discretisation method exploited. Firstly consider a three-dimensional stream function.

Extending the two-dimensional Navier Stokes equations in stream function formulation in to the third dimension is not as immediately obvious as it would be in primitive variable form, which uses velocity and pressure, and makes no assumptions about the number of spatial dimensions. Elashabka and Chung[161] and Chung[99] introduce a methodology for modelling a three-dimensional stream function. Here two stream functions are used visualised as forming ducts along which the mass flux is constant. The ψ-surfaces form the walls (as in two-dimensions) with the addition of χ-surfaces forming the floors and ceilings of the channels. A similar approach is given by Greywall[162].

Topological subdivision is also not as straight forward in three dimensions. On a two-dimensional triangular mesh the subdivision rules, as detailed in chapter five, generate four similar triangles from each triangle in the mesh. This is in contrast to the three-dimensional tetrahedral counterpart as given by Chang *et al.*[61, 62] and Schaefer *et al.*[163]

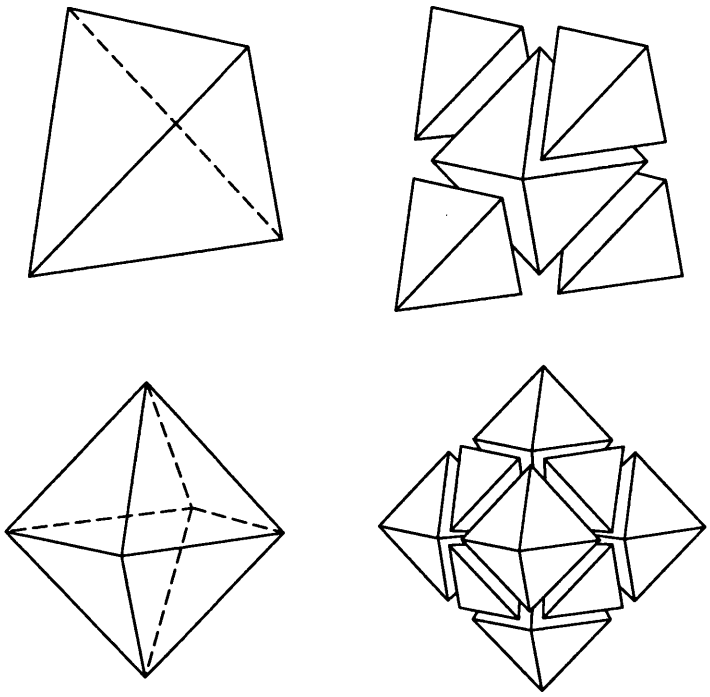


Figure 10.2 Three dimensional subdivision of a tetrahedron. Illustration reproduced from page 229 of Chang *et al.*[61]

Here, as illustrated in Figure 10.2, a tetrahedron is subdivided into four similar tetrahedra, but with an additional central octahedron. Subdividing the octahedron yields six more octahedra with eight tetrahedra. Areas for possible further work may be to extend this as a finite element discretisation of the three-dimensional stream function Navier-Stokes equations.

Finally it should be again noted that the case of high Reynolds number fluid flow is a severe test case with approximations and instabilities being convected throughout the domain and with little damping from the relatively low viscosity. Other possible high order differential equations or structural applications, following on from the work of Cirak *et al.*[84], Dondl *et al.*[93] and Green and Turkiyyah[91], may be less sensitive to changes in the resolution of the discretisation. Implementation of the proposed solution methodology to further test cases therefore constitutes possible further research.

Part four

Appendices

Appendix A

Biomimesis and the Geometric Definition of Shell Structures in Architecture

Biomimesis and the Geometric Definition of Shell Structures in Architecture

Semra Arslan Selcuk
Department of Architecture, Middle East Technical University, Ankara, Turkey.
e-mail: semra@arch.metu.edu.tr

Al J Fisher, MEng
Department of Architecture and Civil Engineering, University of Bath, UK.
e-mail: a.j.fisher@bath.ac.uk

Chris JK Williams, MA, PhD, MIStructE
Department of Architecture and Civil Engineering, University of Bath, UK.
e-mail: c.j.k.williams@bath.ac.uk

Abstract

In architecture the concept of biomimesis (bios, meaning life, mimesis, meaning to imitate, also known as biomimetics) can be applied to the design of tree-like, web-like, skeleton-like, pneumatic and shell-like structures.

Shell structures may be constructed from masonry, concrete or a grid of steel or timber members – a lattice or reticulated shell. The paper examines the philosophical, æsthetic, structural and environmental criteria to be used to decide whether biomimesis is appropriate. A strategy is described for the generation of organic forms in which a three-dimensional computer model is produced using a combination of interactive sculpting and analytical relationships. The inspiration behind this approach is the geometry of seashells.

1. Introduction

A shell is a three dimensional curved structure which resists load through its inherent curvature. There are numerous shell structures in nature; eggs, skulls, nuts, turtles and seashells, are notable examples [1]. A shell's structural behaviour is derived directly from its form, thus when designing a shell-like structure, the fundamental consideration is the choice of geometry. This not only dictates the æsthetics, but the overall efficiency and behaviour under load of the structural system. The basis for curved geometry, as discussed by Williams [2], can be sculptural, geometric or defined by a natural physical process.

There are many precedents of the use of natural forms and phenomena as the basis for architectural geometry. Techniques for forming optimum shapes for pure tensile or

compressive structures have been developed using physical models. Antonio Gaudí aspired to create optimal structural forms using inverted string models, taking his inspiration from Gothic architecture in Spain.
A string, when suspended at each end and allowed to hang freely, forms a catenary, a curve of pure tension. Using this principle Gaudí formed pure compression geometries by inverting the form of the hanging models. Gaudí also experimented with suspending cloth, allowing it to sag to create three-dimensional vault surfaces. Throughout the twentieth century architects and engineers, including Torroja, Nervi, Candella and Isler, explored the design of thin concrete shell structures. Figure 1 shows a service station in Deitingen-Süd, Switzerland, designed by Heinz Isler. Much of Isler's free-form geometry was generated using inverted hanging models similar to Gaudí's. Isler also went on to experiment with pneumatic membrane models, creating what he called 'bubble shells', where the geometry was defined by air inflated membranes [4].

In 1964 Frei Otto founded the Institute of Lightweight Structures at the University of Stuttgart. Here Otto and his team undertook extensive research into natural forms and processes, generating and inspiring optimum structures [5]. As well as shell structures, lightweight or optimum structures include membrane structures, cable-net, geodesic domes, inflatable and air supported structures, and lattice or reticulated shell structures such as the grid shell pictured in figure 2. As well as building on the historical precedents of inverted hanging models for rigid compression structures, Otto pioneered the use of soap film models to generate minimal surfaces defining the optimum geometry of tensile structures (figure 3).

Further examples of biomimesis in architecture may include Calatrava's bone like structures [6], Meinhard von Gerkan's tree-like roof of the Stuttgart Airport passenger terminal or perhaps Nicholas Grimshaw's geodesic domes of the Eden project in Cornwall, which bear a striking resemblance to the hexagonal lattice of a diatom, pictured in figure 3.

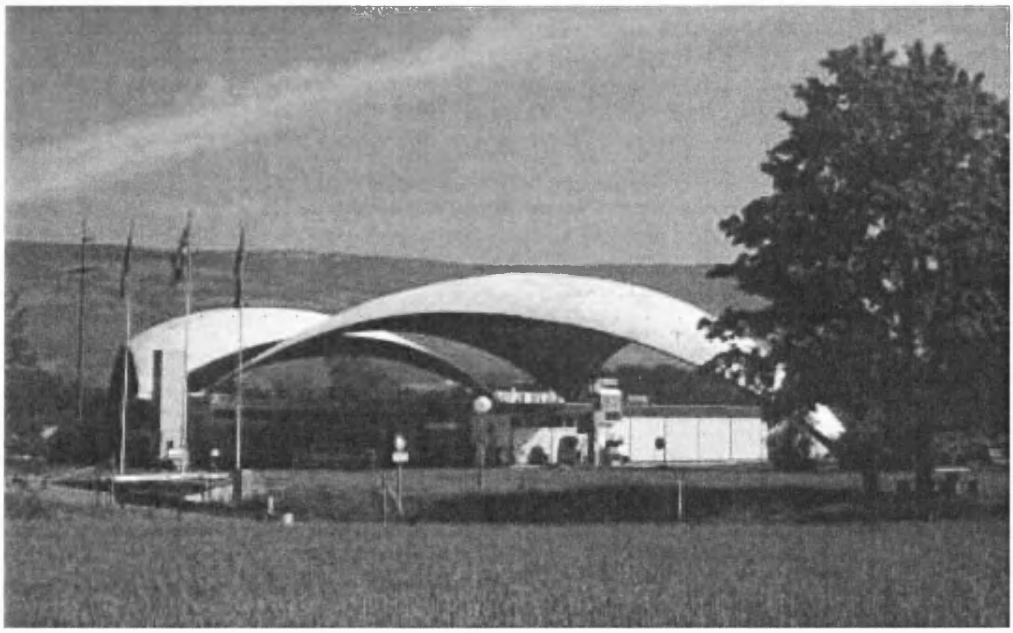


Figure 1: Concrete shell structure designed by Heinz Isler [3]

Figure 2: Grid shell roof covering the Great Court of the British Museum in London Architect: Foster and Partners, Engineer: Buro Happold

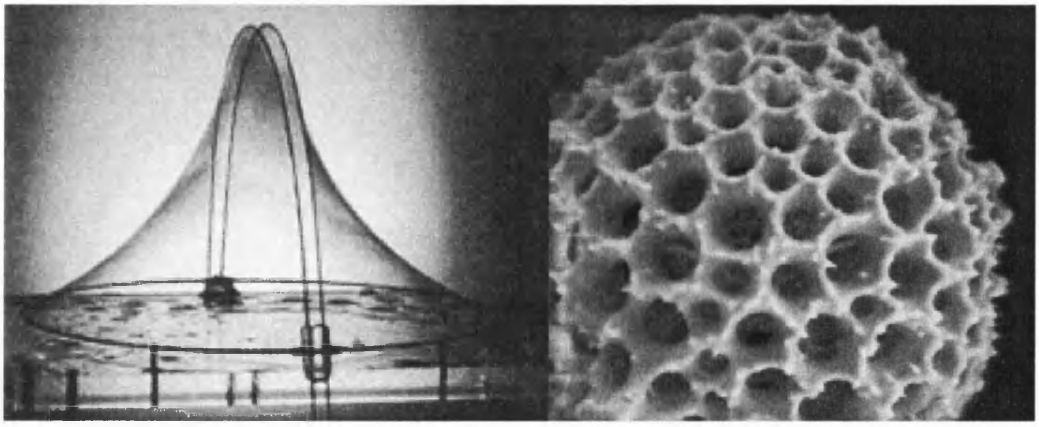


Figure 3: (left) Soap film experiment producing minimal surface [5] (right) Diatom, analogous to a geodesic dome [7]

Although there are numerous examples of where biomimesis has been successfully applied to architecture, the potential is far from being exhausted. There are endless different structural forms played out in nature, however in most cases the complexity of these forms renders geometric modelling problematic. Yet as computer-aided design and computational analysis techniques are becoming more powerful, greater interest is focused on these forms and even complex structural behaviour is modelled successfully. It is the aim of this paper to describe a method for modelling seashell forms with an application in architectural design.

As described above a shell's inherent shape defines how efficiently loads are resisted. Loads applied to a shell surface can be accommodated through two actions, through bending and through stretching [8]. Bending forces are carried through moments and shear forces, and stretching forces through axial thrusts and tensions. The later of these phenomena is called membrane action. Membrane action is much more efficient than bending, and therefore a shell will work by membrane action if it can, however to determine whether a shell can or not depends greatly upon its geometry and support conditions. Membrane action also relies on bending stiffness to prevent buckling. Therefore in order to understand the structural behaviour of seashells a complete understanding of their geometry is required.

2. Geometry of Seashells

Seashells are formed in nature by growth at the shell's free leading edge. Their increase in overall size is achieved purely from successive addition of material to one end only [9]. From inspection of actual seashell cross-sections, the older previously formed parts of the shell remain, on the whole, unaffected and geometrically unchanged once produced (as illustrated in figure 4).

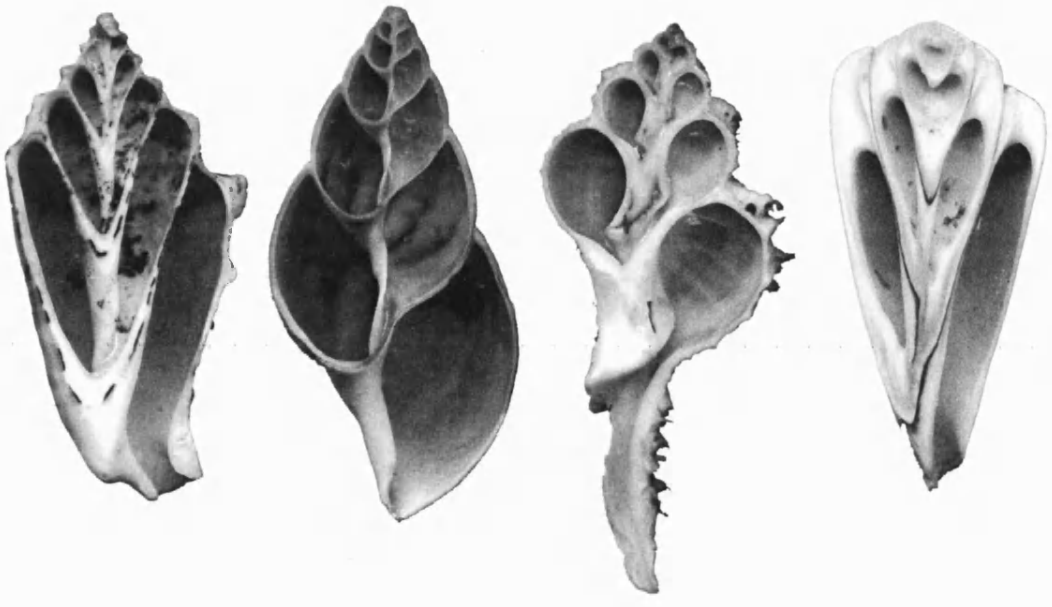


Figure 4: Cut cross-sections of a selection of seashells found in nature

A simple seashell may be considered as a surface of revolution formed along a spiral path about the shell's axis. The generating cross-section is of constant shape, but increasing in size by a constant ratio as the section sweeps the curve. For centuries biologists and mathematicians have explored techniques for describing and dissecting this shell geometry into simple terms.

The origins of theoretical seashell morphology can be traced back to the work of Moseley, who derived equations for calculating the volume, surface area, and centre of gravity of planispiral and trochospiral shells [10]. Moseley's equations were reformulated by Raup with the application of displaying shell shapes using a computer [11]. Raup described the geometry of seashells using three parameters, which he called W, D and T. These terms were later elaborated by Dawkins [12], renaming them flare, verm and spire, respectively.

Flare, Dawkins explains, is a measure of the expansion rate of the spiral, verm controls the rate of increase in size of the cross-section and spire is the rate of displacement along the axis of the shell. Figure 5 illustrates the effect on the overall shell geometry for changes in these different parameters, using an ellipse as the whorl cross-section. However it is clear from observations of actual shells (figure 4) that the cross-section is more complex than the input of three parameters allows. Fowler *et al.* [13] were first to implement free-form cross sections, using a Bézier curve [14,15] as the input.

All these approaches model the shell as a single surface, a two-dimensional object, embedded in three-dimensional space. To the writers' knowledge little work has been done on accurately modelling the cross-section of the shell, showing the thickness of the shell wall and the complex solid volumes that are formed down the internal spine. As described above shells as a structural mechanism are incredibly sensitive to variations in geometry. It is this paper's intent to create a method for generating a three-dimensional shell model which includes the thickness of the shell material, thus providing a model which can form the basis of a three-dimensional structural analysis.

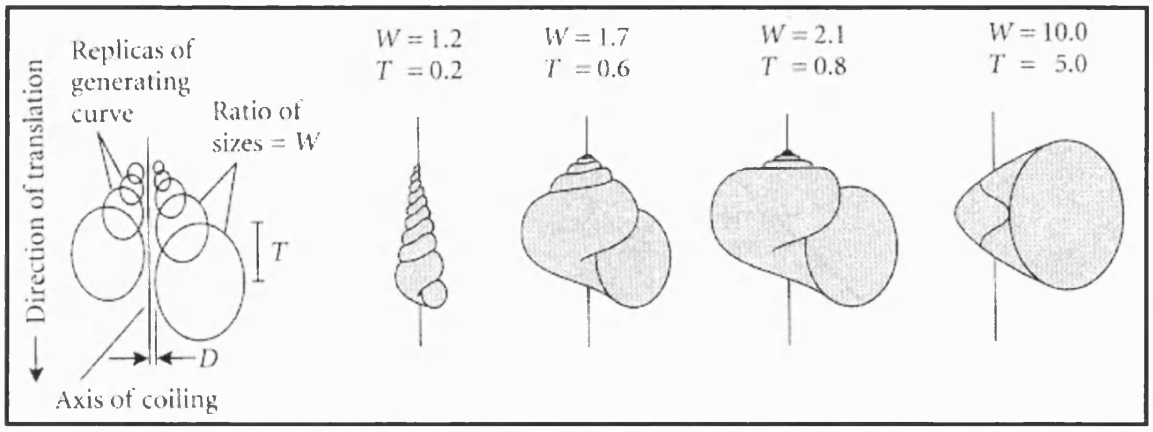


Figure 5: Shell geometry controlled by parameters flare (W), verm (D) and spire (T) [11]

3. Construction of the shell model

The basic approach to modelling the shell's solid cross-sections was to first generate the internal whorl surface which forms the cavity in which the gastropod lives. A portion of this surface could then be offset defining the outermost surface of the shell. The volume between these two surfaces would then form the geometry of the solid material which forms the shell.

A parametric model of the shell was set up using the cross section of a single whorl as the input. In a similar approach to Fowler *et al* [13], the shape of a single typical whorl was defined as a B-spline. A B-spline can be used to approximate a smooth curve from a small set of control points [14,15]. Figure 6 shows a B-spline of order four, and its control polygon.

From the definition of a B-spline the curve is parameterised along its length, with respect to parameter μ with a range of $0 \leq \mu \leq 1.0$. $\mu = 0$ and 1.0 refer to the start and end of the closed loop respectively whilst the point $\mu = 0.5$ is half way around the length of the curve.

Between the values $\mu_{begin} = 0$ and μ_{join} the whorl B-spline is offset by a given thickness to form the outer surface of the shell. The outer surface is parameterised between ϕ_{begin} and ϕ_{end}. A single whorl cross-section in its local coordinate system is illustrated in figure 6.
The ribs on the external shell surface, as illustrated later in figures 9 and 10, were generated by superimposing a sine based function on to the surface normal component of the position vectors.

The shell surface geometry is defined using cylindrical polar coordinates (r, θ, z), which can be expressed in terms of Cartesian coordinates (x, y, z):

$$\begin{aligned} x &= r\cos\theta \\ y &= r\sin\theta \\ z &= z \end{aligned} \qquad (1)$$

The path along which a shells whorl cross-section follows is a logarithmic spiral and it is the geometric properties of this curve that define the overall geometry of the shell. A logarithmic spiral is a curve which forms a constant angle between its tangent and radius vector at any point. Hence logarithmic spirals have the alternative name *equiangular spirals*. Figure 7 illustrates the constant λ which defines the rate of spiral for any such curve.

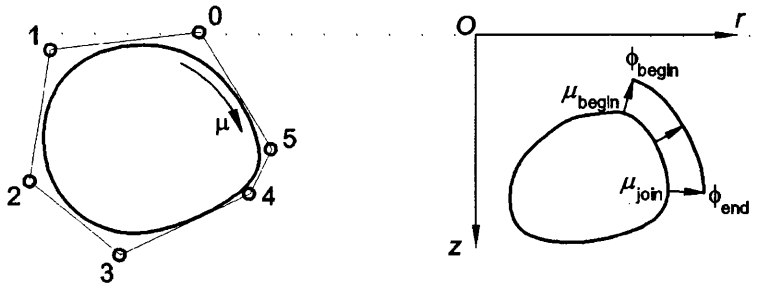


Figure 6: (left) B-spline and its control polygon
(right) single whorl cross-section

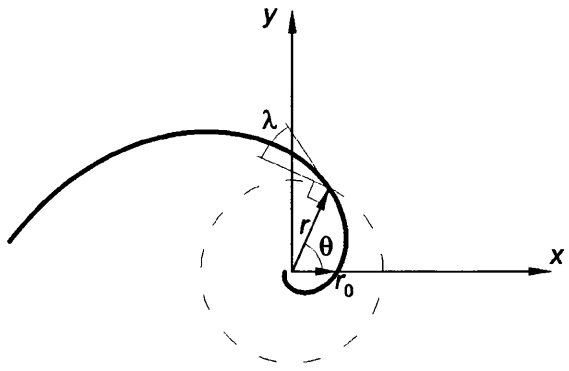


Figure 7: Geometry of the logarithmic spiral.

The relationship between any two points on a logarithmic spiral can be described by the formula below:

$$\tan\lambda = \frac{1}{r}\frac{dr}{d\theta} \tag{2}$$

giving

$$e^{\theta \tan\lambda} = \frac{r}{r_0} \tag{3}$$

where r_0 is the radius at $\theta = 0$.

From observations made from real seashells the growth rings, which correspond to the whorl cross-section, are not radial and do not even lie in one plane. They are curves in three dimensions. This means that the shell's rate of spiral, λ, as previously illustrated, is required to be such that the point ϕ_{begin} on the current leading edge cross-section must lie coincident with ϕ_{end} on the preceding section after slightly less than one revolution about the major z-axis, i.e. $\Delta\theta = 2\pi - \theta_{join}$, as illustrated in figure 8 and expressed in the following equation (4):

$$r(\theta, \phi_{end}) = r(\theta + 2\pi - \theta_{join}, \phi_{begin}) \tag{4}$$

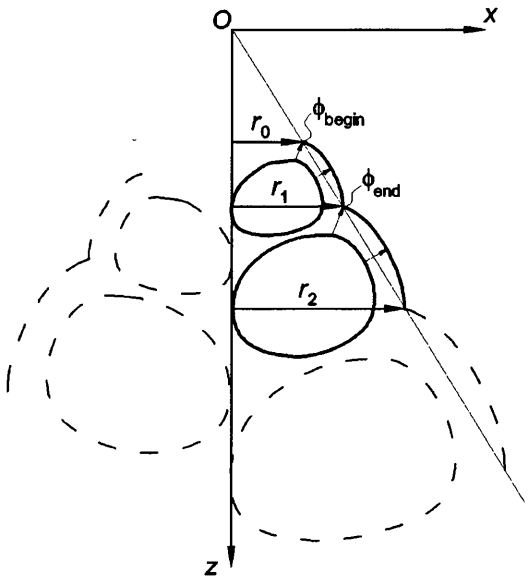


Figure 8: Geometric relations of the shell surface

Thus λ is controlled by the relationship:

$$e^{(2\pi-\theta_{join})\tan\lambda} = \frac{r_1}{r_0} \tag{5}$$

where:

λ = rate of spiral constant

r_0 = radius to point ϕ_0

r_1 = radius to point ϕ_0'

$\theta_{join} = \theta_{shift}$ at point ϕ_{join}

The growth constant is applied to the whorl by transforming its coordinates using the formulae below, based on the cylindrical polar coordinate transformation (equation (1)):

$$\begin{aligned} x &= e^{\theta\tan\lambda} r\cos\theta \\ y &= e^{\theta\tan\lambda} r\sin\theta \\ z &= e^{\theta\tan\lambda} z \end{aligned} \tag{6}$$

Figure 8 illustrates the sequential increase in size of the cross-section, starting from an infinitesimal size at the origin.

The shape of the whorl, the value of ϕ_{end} and the offset thickness are all parameters easily measured from actual seashell cross-sections and it is from these inputs that the whole three-dimensional shell geometry can be generated. Figure 9 illustrates a good correlation between the real natural seashell on the left and the computer generated model on the right.

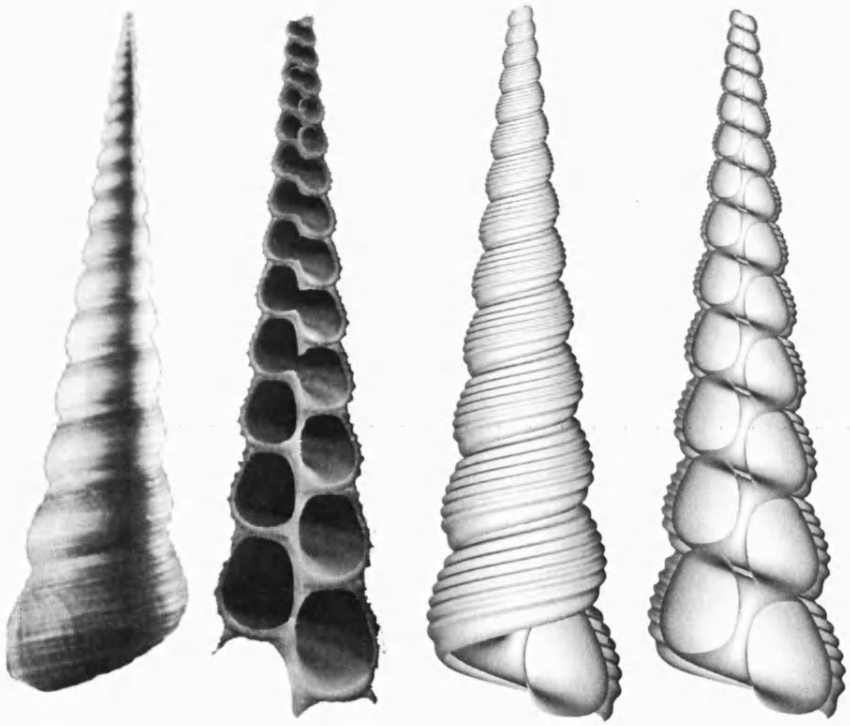


Figure 9: (left) scan of a seashell found in nature
(right) computer generated shell model

4. Conclusions

Seashells were investigated as three-dimensional structures as part of a study into biomimesis as a basis for architectural form. Previous seashell models have accurately captured the external surface of the shells, generating realistic images, however our research has placed emphasis on the shell as a volume.
Figure 10 shows a three-dimensional model which has been clipped to reveal the central third of the shell, illustrating the incredibly complex volumes and voids which are generated internally.
The next phase of the research is to understand the structural implications of this geometry and thus suggest a set of inspired architectural forms.

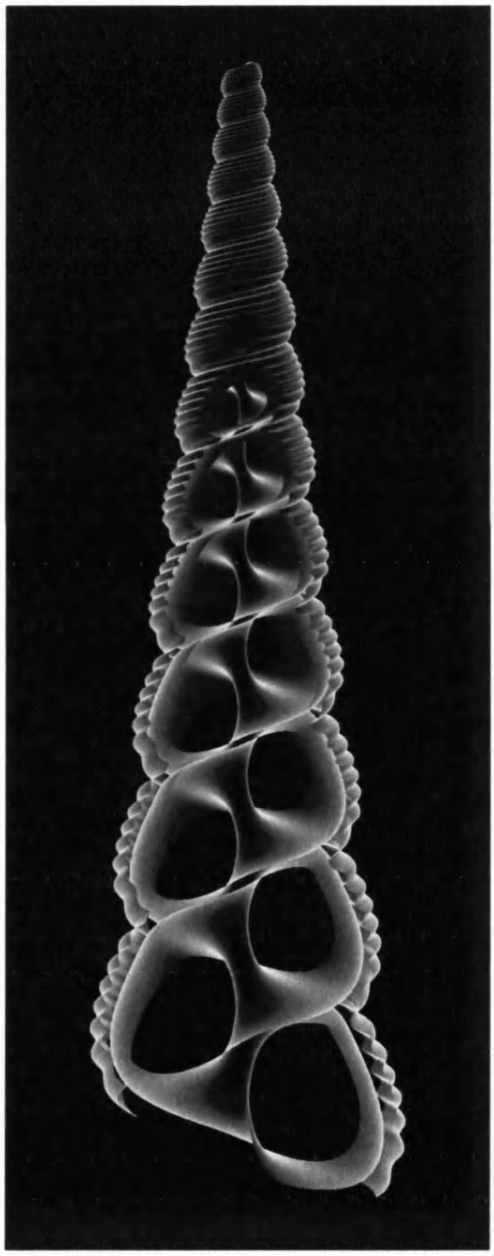


Figure 10: Three-dimensional shell model with two cutting planes at third points

References

[1] Arslan, S. and Gonenc Sorguc, A., 2004, Similarities in Structures in Nature and Man-Made Structures: Biomimesis in Architecture. Proceedings of the *2nd Design and Nature Conference Comparing Design in Nature with Science and Engineering,* Rhodes, Greece, June 28th - 30th, 2004.

[2] Williams, C.J.K., 2000, The definition of curved geometry for widespan enclosures, Proceedings of the *International Symposium on widespan enclosures,* University of Bath, UK, 26th – 28th April 2000.

[3] Heinz Isler [Online] Available http://www.anc-d.fukui-u.ac.jp/~ishikawa/Aloss/data/Isler/Catalog/Isler_catalog_files/main_catalog.html (Accessed 3rd November 2005)

[4] Lewis, W., 2005, Understanding novel structures through form-finding. *Proceedings of the Institution of Civil Engineers,* 158(4), pp. 178-185.

[5] Otto, F., 1995, *Frei Otto, Bodo Rasch: Finding Form. Towards an Architecture of the Minimal*. Deutscher Werkbund Bayern.

[6] Zardini, M., 1996, *Santiago Calatrava Secret Sketchbook.* New York: Monacelli Press.

[7] Northwestern University Atomic and Nanoscale Characterization Experiment Centre [Online] Available at http://www.nuance.northwestern.edu/ (Accessed 3rd November 2005)

[8] Calladine, C.R., 1983, *Theory of shell structures.* Cambridge: University Press.

[9] D'Arcy Thompson, 1992, *On Growth and Form.* Cambridge: University Press.

[10] Moseley, H., 1838, On the geometrical forms of turbinated and discoid shells. *Philosophical Transactions of the Royal Society of London,* pp. 351-370.

[11] Raup, D.M., 1961, The Geometry of Coiling in Gastropods, Proceedings of the *National Academy of Sciences of the United States of America,* Vol. 47, pp. 602–609.

[12] Dawkins, R., 1997, *Climbing Mount Improbable.* W.W. Norton & Company.

[13] Fowler, D.R., Meinhardt, H, and Prusinkiewicz, P., 1992. Modeling Seashells. *Computer graphics,* 26(2), pp. 379-387.

[14] Farin, G., 2002, *Curves and surfaces for CAGD: a pratical guide.* 5th ed. San Franciso: Morgan Kaufmann Publishers.

[15] Rogers, D.F., 2001, *An introduction to NURBS with historical perspective.* Morgan Kaufmann Publishers.

Appendix B

Subdivision shape functions and their first and second derivatives

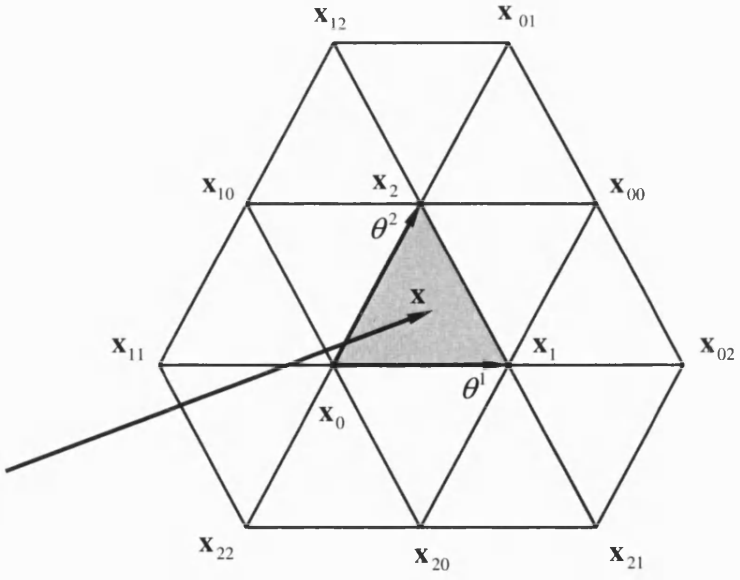


Figure B.1 Twelve control points governing a regular subdivision patch

Following Stam[86], whilst using the numbering convention given in Figure B.1, the twelve shape functions are given as

$$N^0 = \frac{1}{12}\left(6u^4 + 24u^3w + 24u^2w^2 + 8uw^3 + w^4 + 24u^3v + 60u^2vw + 36uvw^2 + 6vw^3 + 24u^2v^2 + 36uv^2w + 12v^2w^2 + 8uv^3 + 6v^3w + v^4\right),$$

$$N^{00} = \frac{1}{12}\left(2uw^3 + w^4 + 6uvw^2 + 6vw^3 + 6uv^2w + 12v^2w^2 + 2uv^3 + 6v^3w + v^4\right),$$

$$N^{01} = \frac{1}{12}\left(2v^3w + v^4\right),$$

$$N^{02} = \frac{1}{12}\left(w^4 + 2vw^3\right),$$

$$N^1 = \frac{1}{12}\left(u^4 + 6u^3w + 12u^2w^2 + 6uw^3 + w^4 + 8u^3v + 36u^2vw + 36uvw^2 + 8vw^3 + 24u^2v^2 + 60uv^2w + 24v^2w^2 + 24uv^3 + 24v^3w + 6v^4\right),$$

$$N^{10} = \frac{1}{12}\left(u^4 + 6u^3w + 12u^2w^2 + 6uw^3 + w^4 + 2u^3v + 6u^2vw + 6uvw^2 + 2vw^3\right),$$

$$N^{11} = \frac{1}{12}\left(2uw^3 + w^4\right),$$

$$N^{12} = \frac{1}{12}\left(u^4 + 2u^3w\right),$$

$$N^2 = \frac{1}{12}\left(u^4 + 8u^3w + 24u^2w^2 + 24uw^3 + 6w^4 + 6u^3v + 36u^2vw + 60uvw^2 + 24vw^3 + 12u^2v^2 + 36uv^2w + 24v^2w^2 + 6uv^3 + 8v^3w + v^4\right),$$

$$N^{20} = \frac{1}{12}\left(u^4 + 2u^3w + 6u^3v + 6u^2vw + 12u^2v^2 + 6uv^2w + 6uv^3 + 2v^3w + v^4\right),$$

$$N^{21} = \frac{1}{12}\left(u^4 + 2u^3v\right),$$

$$N^{22} = \frac{1}{12}\left(2uv^3 + v^4\right).$$

All superscripts on the right-hand side of the equations represent exponents and the curvilinear coordinates θ^1 and θ^2 conform to the following expressions

$$\begin{aligned} v &= \theta^1, \\ w &= \theta^2, \\ u &= 1 - \theta^1 - \theta^2. \end{aligned}$$

B.1 First derivatives of the shape function

$$N^{0}{}_{,1} = \frac{1}{12}\left(-12u^2w - 12uw^2 - 2w^3 - 24u^2v - 48uvw - 12vw^2 - 24uv^2 - 18v^2w - 4v^3\right),$$

$$N^{00}{}_{,1} = \frac{1}{12}\left(6uw^2 + 4w^3 + 12uvw + 18vw^2 + 2v^3 + 6uv^2 + 12v^2w\right),$$

$$N^{01}{}_{,1} = \frac{1}{12}\left(6v^2w + 4v^3\right),$$

$$N^{02}{}_{,1} = \frac{1}{12}\left(2w^3\right).$$

$$N^{1}{}_{,1} = \frac{1}{12}\left(12v^2w + 12vw^2 + 2w^3 + 24uv^2 + 48uvw + 12uw^2 + 24u^2v + 18u^2w + 4u^3\right),$$

$$N^{10}{}_{,1} = \frac{1}{12}\left(-6vw^2 - 4w^3 - 12uvw - 18uw^2 - 2u^3 - 6u^2v - 12u^2w\right),$$

$$N^{11}{}_{,1} = \frac{1}{12}\left(-2w^3\right),$$

$$N^{12}{}_{,1} = \frac{1}{12}\left(-6u^2w - 4u^3\right).$$

$$N^{2}{}_{,1} = \frac{1}{12}\left(2u^3 + 6u^2v + 12u^2w + 12uw^2 - 2v^3 - 6uv^2 - 12v^2w - 12vw^2\right),$$

$$N^{20}{}_{,1} = \frac{1}{12}\left(2u^3 + 6u^2v - 6uv^2 - 2v^3\right),$$

$$N^{21}{}_{,1} = \frac{1}{12}\left(-2u^3 - 6u^2v\right),$$

$$N^{22}{}_{,1} = \frac{1}{12}\left(2v^3 + 6uv^2\right).$$

$$N^{0}{}_{,2} = \frac{1}{12}\left(-12u^2v - 12uv^2 - 2v^3 - 24u^2w - 48uvw - 12v^2w - 24uw^2 - 18vw^2 - 4w^3\right),$$

$$N^{00}{}_{,2} = \frac{1}{12}\left(6uv^2 + 4v^3 + 12uvw + 18v^2w + 2w^3 + 6uw^2 + 12vw^2\right),$$

$$N^{01}{}_{,2} = \frac{1}{12}\left(2v^3\right),$$

$$N^{02}{}_{,2} = \frac{1}{12}\left(6vw^2 + 4w^3\right).$$

$$N^{1}{}_{,2} = \frac{1}{12}\left(2u^3 + 6u^2w + 12u^2v + 12uv^2 - 2w^3 - 6uw^2 - 12vw^2 - 12v^2w\right),$$

$$N^{10}{}_{,2} = \frac{1}{12}\left(2u^3 + 6u^2w - 6uw^2 - 2w^3\right),$$

$$N^{11}{}_{,2} = \frac{1}{12}\left(2w^3 + 6uw^2\right),$$

$$N^{12}{}_{,2} = \frac{1}{12}\left(-2u^3 - 6u^2w\right).$$

$$N^{2}{}_{,2} = \frac{1}{12}\left(12vw^2 + 12v^2w + 2v^3 + 24uw^2 + 48uvw + 12uv^2 + 24u^2w + 18u^2v + 4u^3\right),$$

$$N^{20}{}_{,2} = \frac{1}{12}\left(-6v^2w - 4v^3 - 12uvw - 18uv^2 - 2u^3 - 6u^2w - 12u^2v\right),$$

$$N^{21}{}_{,2} = \frac{1}{12}\left(-6u^2v - 4u^3\right),$$

$$N^{22}{}_{,2} = \frac{1}{12}\left(-2v^3\right).$$

B.2 Second derivatives of the shape function

$$N^{0},_{11} = \frac{1}{12}\left(12vw + 12v^2 - 24uv - 24u^2\right),$$

$$N^{00},_{11} = \frac{1}{12}\left(12w^2 + 12vw + 12uw + 12uv\right),$$

$$N^{01},_{11} = \frac{1}{12}\left(12vw + 12v^2\right),$$

$$N^{02},_{11} = 0.$$

$$N^{1},_{11} = \frac{1}{12}\left(12uw + 12u^2 - 24vw - 24v^2\right),$$

$$N^{10},_{11} = \frac{1}{12}\left(12w^2 + 12uw + 12vw + 12uv\right),$$

$$N^{11},_{11} = 0,$$

$$N^{12},_{11} = \frac{1}{12}\left(12uw + 12w^2\right).$$

$$N^{2},_{11} = \frac{1}{12}\left(-24uv - 24uw - 24vw - 24w^2\right),$$

$$N^{20},_{11} = \frac{1}{12}\left(-24uv\right),$$

$$N^{21},_{11} = \frac{1}{12}\left(12uv\right),$$

$$N^{22},_{11} = \frac{1}{12}\left(12uv\right).$$

$$N^{0}{}_{,12} = \frac{1}{12}\left(6w^2 + 24vw + 6v^2 - 12u^2\right),$$

$$N^{00}{}_{,12} = \frac{1}{12}\left(6w^2 + 12uw + 24vw + 12uv + 6v^2\right),$$

$$N^{01}{}_{,12} = \frac{1}{12}\left(6v^2\right),$$

$$N^{02}{}_{,12} = \frac{1}{12}\left(6w^2\right).$$

$$N^{1}{}_{,12} = \frac{1}{12}\left(6u^2 - 12v^2 - 24vw - 6w^2 - 12uw\right),$$

$$N^{10}{}_{,12} = \frac{1}{12}\left(6w^2 - 12uw - 6u^2\right),$$

$$N^{11}{}_{,12} = \frac{1}{12}\left(-6w^2\right),$$

$$N^{12}{}_{,12} = \frac{1}{12}\left(12uw + 6u^2\right).$$

$$N^{2}{}_{,12} = \frac{1}{12}\left(6u^2 - 12w^2 - 24vw - 6v^2 - 12uv\right),$$

$$N^{20}{}_{,12} = \frac{1}{12}\left(6v^2 - 12uv - 6u^2\right),$$

$$N^{21}{}_{,12} = \frac{1}{12}\left(12uv + 6u^2\right),$$

$$N^{22}{}_{,12} = \frac{1}{12}\left(-6v^2\right).$$

$$N^{0}{}_{,22} = \frac{1}{12}\left(12vw + 12w^2 - 24uv - 24u^2\right),$$

$$N^{00}{}_{,22} = \frac{1}{12}\left(12v^2 + 12vw + 12uv + 12uw\right),$$

$$N^{01}{}_{,22} = 0,$$

$$N^{02}{}_{,22} = \frac{1}{12}\left(12vw + 12w^2\right).$$

$$N^{1}{}_{,22} = \frac{1}{12}\left(-24uw - 24uv - 24vw - 24v^2\right),$$

$$N^{10}{}_{,22} = \frac{1}{12}\left(-24uw\right),$$

$$N^{11}{}_{,22} = \frac{1}{12}\left(12uw\right),$$

$$N^{12}{}_{,22} = \frac{1}{12}\left(12uw\right).$$

$$N^{2}{}_{,22} = \frac{1}{12}\left(12uv + 12u^2 - 24vw - 24w^2\right),$$

$$N^{20}{}_{,22} = \frac{1}{12}\left(12v^2 + 12uv + 12vw + 12uw\right),$$

$$N^{21}{}_{,22} = \frac{1}{12}\left(12uv + 12u^2\right),$$

$$N^{22}{}_{,22} = 0.$$

Appendix C

Navier solution for simply supported rectangular plate under uniformly distributed loading

As given by Jaeger[143], Lowe[144] and Timoshenko and Woinowsk-Krieger[145] among others, assume sinusoidal loading of the form

$$p = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} q_{mn} \sin\frac{m\pi x}{a}\sin\frac{n\pi y}{b}, \qquad \text{[C.1]}$$

where p is per unit area.

Multiplying both sides by $\sin\frac{m'\pi x}{a}dx$ and integrating from $x=0$ to $x=a$ we obtain for the left-hand side of the equation

$$\int_0^a p\sin\frac{m'\pi x}{a}dx = -p\left[\frac{a}{m'\pi}\cos\frac{m'\pi x}{a}\right]_0^a$$
$$= \frac{2pa}{m'\pi} \text{ if } m' \text{ is odd}$$
$$= 0 \text{ if } m' \text{ is even.}$$

and the right-hand side

$$\int_0^a \sin\frac{m\pi x}{a}\sin\frac{m'\pi x}{a}dx = 0 \text{ when } m \neq m'$$

$$= \frac{a}{2} \text{ when } m = m'.$$

therefore

$$\frac{2pa}{m'\pi} = \frac{a}{2}\sum_{n=1}^{\infty} q_{m'n}\sin\frac{n\pi y}{b}.$$

Similarly, now multiplying both sides by $\sin\frac{n'\pi y}{b}dy$ and integrating from $y = 0$ to $y = b$, we obtain

$$\frac{4pab}{m'n'\pi^2} = \frac{ab}{4}q_{m'n'}, \qquad \text{[C.2]}$$

giving an expression for any particular coefficient as

$$q_{m'n'} = \frac{16p}{m'n'\pi^2}. \qquad \text{[C.3]}$$

The required boundary conditions for simply supported edges are,

$$\psi(0,y) = 0,$$
$$\psi(a,y) = 0,$$
$$\psi(x,0) = 0,$$
$$\psi(x,b) = 0,$$
$$\frac{\partial^2\psi}{\partial x^2}(0,y) = 0,$$
$$\frac{\partial^2\psi}{\partial x^2}(a,y) = 0,$$
$$\frac{\partial^2\psi}{\partial y^2}(x,0) = 0 \text{ and}$$
$$\frac{\partial^2\psi}{\partial y^2}(x,b) = 0,$$

and the additional requirements that

$$\frac{\partial \psi}{\partial x}(x,0)=0,$$
$$\frac{\partial \psi}{\partial x}(x,b)=0, \text{ but}$$
$$\frac{\partial \psi}{\partial x}(0,y)\neq 0,$$
$$\frac{\partial \psi}{\partial x}(a,y)\neq 0 \text{ and}$$
$$\frac{\partial \psi}{\partial y}(0,y)=0,$$
$$\frac{\partial \psi}{\partial y}(a,y)=0, \text{ but}$$
$$\frac{\partial \psi}{\partial y}(x,0)\neq 0,$$
$$\frac{\partial \psi}{\partial y}(x,b)\neq 0,$$

are all satisfied if we propose a periodic solution of the form

$$\psi = C\sin\frac{m\pi x}{a}\sin\frac{n\pi y}{b}. \qquad [C.4]$$

Substituting into the plate equation in Cartesian form

$$\nabla^4\psi = \frac{\partial^4\psi}{\partial x^4}+\frac{2\partial^4\psi}{\partial x^2\partial y^2}+\frac{\partial^4\psi}{\partial y^4}=\frac{p}{D},$$

we have

$$C\frac{m^4\pi^4}{a^4}\sin\frac{m\pi x}{a}\sin\frac{n\pi y}{b}+2C\frac{m^2n^2\pi^4}{a^2b^2}\sin\frac{m\pi x}{a}\sin\frac{n\pi y}{b}+$$
$$C\frac{n^4\pi^4}{b^4}\sin\frac{m\pi x}{a}\sin\frac{n\pi y}{b}=\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\frac{q_{mn}}{D}\sin\frac{m\pi x}{a}\sin\frac{n\pi y}{b}. \qquad [C.5]$$

This can be solved for C to give

$$C = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \frac{q_{mn}}{D\pi^4\left[\frac{m^4}{a^4} + \frac{2m^2n^2}{a^2b^2} + \frac{n^4}{b^4}\right]}. \qquad [C.6]$$

Substituting for C and q_{mn} in our expression for deflection we have

$$\psi = \frac{16p}{D\pi^6}\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\left(\frac{\sin\frac{m\pi x}{a}\sin\frac{n\pi y}{b}}{mn\left[\frac{m^2}{a^2} + \frac{n^2}{b^2}\right]^2}\right). \text{ odd values of } m \text{ and } n. \qquad [C.7]$$

Appendix D

Analytic solution for laminar channel flow

The stream function equation in tensor notation, as given in chapter four, is

$$\left(\frac{\partial}{\partial t}+\varepsilon^{\gamma\eta}\nabla_{\eta}\psi\nabla_{\gamma}-\frac{\mu}{\rho}a^{\gamma\eta}\nabla_{\gamma\eta}\right)\left(a^{\alpha\beta}\nabla_{\alpha\beta}\psi\right)=0\,. \qquad [D.1]$$

For axial laminar channel flow the first two terms are exactly zero, since fully developed flow is independent of time t, and as a result of $v_y=0$ the $\varepsilon^{\gamma\eta}$ means that the second term must be zero.

Similar derivations of the velocity profile can be found in the literature, for example the work of Hu[146] page 121 and White[8] page 114.

Assuming axial laminar flow, in Cartesian coordinates we have

$$v_y = 0,$$

thus the continuity equation reduces to

$$\frac{\partial v_x}{\partial x} = 0, \qquad \text{[D.2]}$$

such that the velocity is independent of x. Thus for fully developed flow independent of time t, the momentum equation in component form reduces

$$\begin{aligned} 0 &= -\frac{\partial p}{\partial x} + \mu \frac{\partial^2 v_x}{\partial y^2}, \\ 0 &= -\frac{\partial p}{\partial y}. \end{aligned} \qquad \text{[D.3]}$$

Denoting the pressure gradient,

$$-\frac{\partial p}{\partial x} = P, \qquad \text{[D.4]}$$

and integrating equation [D.3]

$$v_x = -\frac{P}{2\mu} y^2 + Cy + K. \qquad \text{[D.5]}$$

Applying the no-slip condition along the boundary, that is $v_x = 0$ when $y = \pm b$, equation [D.5] can be solved to give

$$v_x = \frac{P}{2\mu}\left(b^2 - y^2\right). \qquad \text{[D.6]}$$

Finally maximum velocity occurs along the central axis of the duct at $y = 0$, the axis equidistant from both boundary walls, thus

$$v_{max} = \frac{b^2}{2\mu} P, \qquad [D.7]$$

giving

$$v_x = v_{max}\left(1 - \frac{y^2}{b^2}\right). \qquad [D.8]$$

Thus the velocity profile is parabolic. From the definition of the stream function

$$\frac{\partial \psi}{\partial y} = v_x,$$
$$-\frac{\partial \psi}{\partial x} = v_y,$$

calculation of the stream function for laminar channel flow is obtained simply through integration

$$\psi = \int v_x dy - \int v_y dx. \qquad [D.9]$$

Thus from equation [D.8]

$$\psi = v_{max}\left(y - \frac{y^3}{3b^2}\right) + C. \qquad [D.10]$$

Defining a datum, the value of the stream function can be arbitrary imposed at a point in the domain, in this case

$$\psi = 0 \text{ when } y = 0,$$

therefore

$$\psi = v_{\max}\left(y - \frac{y^3}{3b^2} \right). \qquad [D.11]$$

The stream function is therefore described by a cubic with tangency being satisfied at the boundary walls, $y = \pm b$. Also we can obviously note that, similar to the velocity, ψ is independent of x.

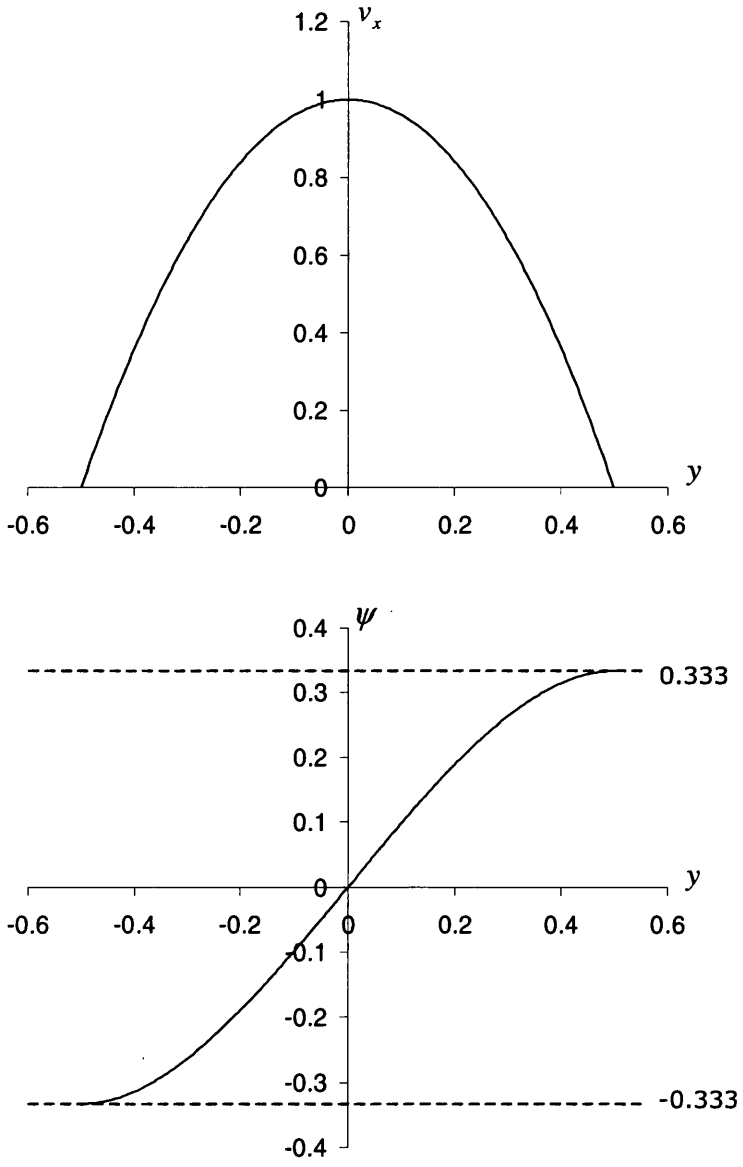


Figure D.1 Velocity and stream function profiles for laminar channel flow of width $2b = 1.0$ and $v_{\max} = 1.0$

The assumptions imposed at the beginning of Appendix D mean that this solution is only stable for low Reynolds numbers and this solution tells us nothing about the limits of this stability.

Appendix E

Analytic solution for plate bending in polar coordinates

The governing equation for plate bending expressed in polar coordinates is given as

$$\nabla^4\psi = \left(\frac{\partial^2}{\partial r^2}+\frac{1}{r}\frac{\partial}{\partial r}+\frac{1}{r^2}\frac{\partial^2}{\partial\theta^2}\right)\left(\frac{\partial^2\psi}{\partial r^2}+\frac{1}{r}\frac{\partial\psi}{\partial r}+\frac{1}{r^2}\frac{\partial^2\psi}{\partial\theta^2}\right)=0, \qquad \text{[E.1]}$$

A solution to this equation needs to satisfy the prescribed boundary conditions. That is,

$$\text{when } r=a,\ w=0 \text{ and } \frac{\partial w}{\partial r}=0.$$

At a radius $r=\eta a$ let us write $w=\eta a\cos\theta$ and $\frac{\partial w}{\partial r}=\cos\theta$.

As $r\to\infty$ we would like $w\to r\cos\theta$, but we shall see later that this is not possible.

Consider solution of the form satisfying the required boundary requirements

$$\psi_n = f_n\left(\frac{r}{a}\right)\cos n\theta . \qquad [E.2]$$

Substitution into the governing equation [E.1] gives

$$\nabla^4\left[f_n\left(\frac{r}{a}\right)\cos n\theta\right] = 0 , \qquad [E.3]$$

which can be expressed in the form

$$\begin{aligned}\nabla^2\left[f_n\left(\frac{r}{a}\right)\cos n\theta\right] &= \left(\frac{\partial^2}{\partial r^2}+\frac{1}{r}\frac{\partial}{\partial r}+\frac{1}{r^2}\frac{\partial^2}{\partial\theta^2}\right)\left[f_n\left(\frac{r}{a}\right)\cos n\theta\right], \\ &= g_n\left(\frac{r}{a}\right)\cos n\theta, \\ \nabla^2\left[g_n\left(\frac{r}{a}\right)\cos n\theta\right] &= \left(\frac{\partial^2}{\partial r^2}+\frac{1}{r}\frac{\partial}{\partial r}+\frac{1}{r^2}\frac{\partial^2}{\partial\theta^2}\right)\left[g_n\left(\frac{r}{a}\right)\cos n\theta\right], \\ &= 0.\end{aligned} \qquad [E.4]$$

The latter part of [E.4] can be written as

$$\frac{1}{a^2}g_n'' + \frac{1}{ra}g_n' - \frac{n^2}{r^2}g_n = 0 . \qquad [E.5]$$

For $n > 0$ equation [E.5] is satisfied by

$$g_n\left(\frac{r}{a}\right) = A_n\left(\frac{r}{a}\right)^n + B_n\left(\frac{r}{a}\right)^{-n}, \qquad [E.6]$$

as shown by substituting [E.6] into [E.5]

$$\frac{1}{a^2}\left(n(n-1)A_n\left(\frac{r}{a}\right)^{n-2}+n(n+1)B_n\left(\frac{r}{a}\right)^{-n-2}\right)$$
$$+\frac{1}{ra}\left(nA_n\left(\frac{r}{a}\right)^{n-1}-nB_n\left(\frac{r}{a}\right)^{-n-1}\right)-\frac{n^2}{r^2}\left(A_n\left(\frac{r}{a}\right)^{n}+B_n\left(\frac{r}{a}\right)^{-n}\right)$$
$$=\left(n^2-n+n-n^2\right)\left(A_n\left(\frac{r}{a}\right)^{n}+B_n\left(\frac{r}{a}\right)^{-n}\right)=0.$$

Alternatively, for $n=0$ equation [E.5] is satisfied by

$$g_0\left(\frac{r}{a}\right)=P\ln\frac{r}{a}+Q. \qquad \text{[E.7]}$$

The solution [E.7] can be seen by expressing equation [E.5] as

$$\frac{a}{r}\left(\frac{r}{a}g_n'\right)'=\frac{n^2}{r^2}g_n.$$

When $n=0$, this simplifies, and by integration it is seen that $\frac{r}{a}g_0'$ is a constant, and hence $g_0'=C\frac{a}{r}$. This leads to the solution [E.7].

Thus we have

$$\frac{f_n''}{a^2}+\frac{1}{ra}f_n'-\frac{n^2}{r^2}f_n=A_n\left(\frac{r}{a}\right)^{n}+B_n\left(\frac{r}{a}\right)^{-n} \text{ if } n>0 \text{ and}$$

$$\frac{f_0''}{a^2}+\frac{1}{ra}f_0'=P\ln\frac{r}{a}+Q \text{ if } n=0.$$

Hence, similarly to that described above, for the case of $n>0$

$$f_n = \frac{A_n\left(\frac{r}{a}\right)^{n+2}}{(n+2)(n+1)+(n+2)-n^2} + \frac{B_n\left(\frac{r}{a}\right)^{2-n}}{(2-n)(1-n)+(2-n)-n^2} + C_n\left(\frac{r}{a}\right)^n + D_n\left(\frac{r}{a}\right)^{-n},$$
$$= E_n\left(\frac{r}{a}\right)^{n+2} + F_n\left(\frac{r}{a}\right)^{2-n} + C_n\left(\frac{r}{a}\right)^n + D_n\left(\frac{r}{a}\right)^{-n}.$$

However for the case of $n=1$ part of the solution is lost and thus a fourth coefficient is required. Thus instead

$$f_1 = E_1\left(\frac{r}{a}\right)^3 + S\frac{r}{a}\log\frac{r}{a} + C_1\left(\frac{r}{a}\right) + D_1\left(\frac{r}{a}\right)^{-1}, \qquad \text{[E.8]}$$

since

$$\frac{1}{a^2}\left(\frac{r}{a}\log\frac{r}{a}\right)'' + \frac{1}{ra}\left(\frac{r}{a}\log\frac{r}{a}\right)' - \frac{1}{r^2}\left(\frac{r}{a}\log\frac{r}{a}\right)$$
$$= \frac{1}{a^2}\frac{a}{r} + \frac{1}{ra}\left(\log\frac{r}{a}+1\right)_n' - \frac{1}{r^2}\left(\frac{r}{a}\log\frac{r}{a}\right) = \frac{2}{ra},$$

satisfying [E.6].

Also, from equation [E.8],

$$a\frac{df_1}{dr} = 3E_1\left(\frac{r}{a}\right)^2 + S\left(\log\frac{r}{a}+1\right) + C_1 - D_1\left(\frac{r}{a}\right)^{-2}.$$

Our boundary conditions mean that the case $n=1$ is sufficient.

Since $w=0$ and $\frac{\partial w}{\partial r}=0$ when $r=a$, from [E.8]

$$E_1 + C_1 + D_1 = 0,$$
$$3E_1 + S + C_1 - D_1 = 0.$$

Thus

$$C_1 = -\left(2E_1 + \frac{S}{2}\right),$$
$$D_1 = \left(E_1 + \frac{S}{2}\right),$$

so that substituting into [E.8]

$$f_1 = E_1\left(\frac{r^3}{a^3} - 2\frac{r}{a} + \frac{a}{r}\right) + S\left(\frac{r}{a}\log\frac{r}{a} - \frac{1}{2}\frac{r}{a} + \frac{1}{2}\frac{a}{r}\right),$$
$$a\frac{df_1}{dr} = E_1\left(3\frac{r^2}{a^2} - 2 - \frac{a^2}{r^2}\right) + S\left(\log\frac{r}{a} + 1 - \frac{1}{2} - \frac{1}{2}\frac{a^2}{r^2}\right). \qquad [E.9]$$

At a radius $r = \eta a$ let us write $w = \eta a\cos\theta$ and $\frac{\partial w}{\partial r} = \cos\theta.$

Therefore

$$\eta a = E_1\left(\eta^3 - 2\eta + \frac{1}{\eta}\right) + S\left(\eta\log\eta - \frac{1}{2}\eta + \frac{1}{2\eta}\right),$$
$$a = E_1\left(3\eta^2 - 2 - \frac{1}{\eta^2}\right) + S\left(\log\eta + \frac{1}{2} - \frac{1}{2\eta^2}\right),$$

which can be rewritten as

$$\eta a = \frac{E_1}{\eta}(\eta^2 - 1)^2 + S\left(\eta\log\eta - \frac{1}{2\eta}(\eta^2 - 1)\right), \qquad [E.10]$$

$$\eta a = \frac{E_1}{\eta}(3\eta^2 + 1)(\eta^2 - 1) + S\left(\eta\log\eta + \frac{1}{2\eta}(\eta^2 - 1)\right). \qquad [E.11]$$

Equating the above we obtain

$$0=\frac{E_1}{\eta}(2\eta^2+2)(\eta^2-1)+S\frac{1}{\eta}(\eta^2-1),$$

and

$$S=-2E_1(\eta^2+1).$$

Hence substituting this expression for S into [E.10]

$$\begin{aligned}\eta a &= E_1\left[\frac{1}{\eta}(\eta^2-1)^2-2(\eta^2+1)\eta\log\eta+\frac{1}{\eta}(\eta^2+1)(\eta^2-1)\right],\\ &= 2\eta E_1[(\eta^2-1)-(\eta^2+1)\log\eta],\end{aligned}$$

so that

$$\begin{aligned}E_1&=\frac{\frac{a}{2}}{(\eta^2-1)-(\eta^2+1)\log\eta},\\ S&=-\frac{(\eta^2+1)a}{(\eta^2-1)-(\eta^2+1)\log\eta},\end{aligned}$$

so, finally, substituting these expressions for E_1 and S into [E.8] we obtain the solution

$$\frac{f_1}{a}=\frac{\frac{1}{2}\left(\frac{r^3}{a^3}-2\frac{r}{a}+\frac{a}{r}\right)-(\eta^2+1)\left(\frac{r}{a}\log\frac{r}{a}-\frac{1}{2}\frac{r}{a}+\frac{1}{2}\frac{a}{r}\right)}{(\eta^2-1)-(\eta^2+1)\log\eta}, \qquad [E.12]$$

$$\frac{df_1}{dr}=\frac{\frac{1}{2}\left(3\frac{r^2}{a^2}-2-\frac{a^2}{r^2}\right)-(\eta^2+1)\left(\log\frac{r}{a}+\frac{1}{2}-\frac{1}{2}\frac{a^2}{r^2}\right)}{(\eta^2-1)-(\eta^2+1)\log\eta}. \qquad [E.13]$$

Stokes' paradox is thus explained by considering equation [E.10] as $\eta\to\infty$, see Figure E.1.

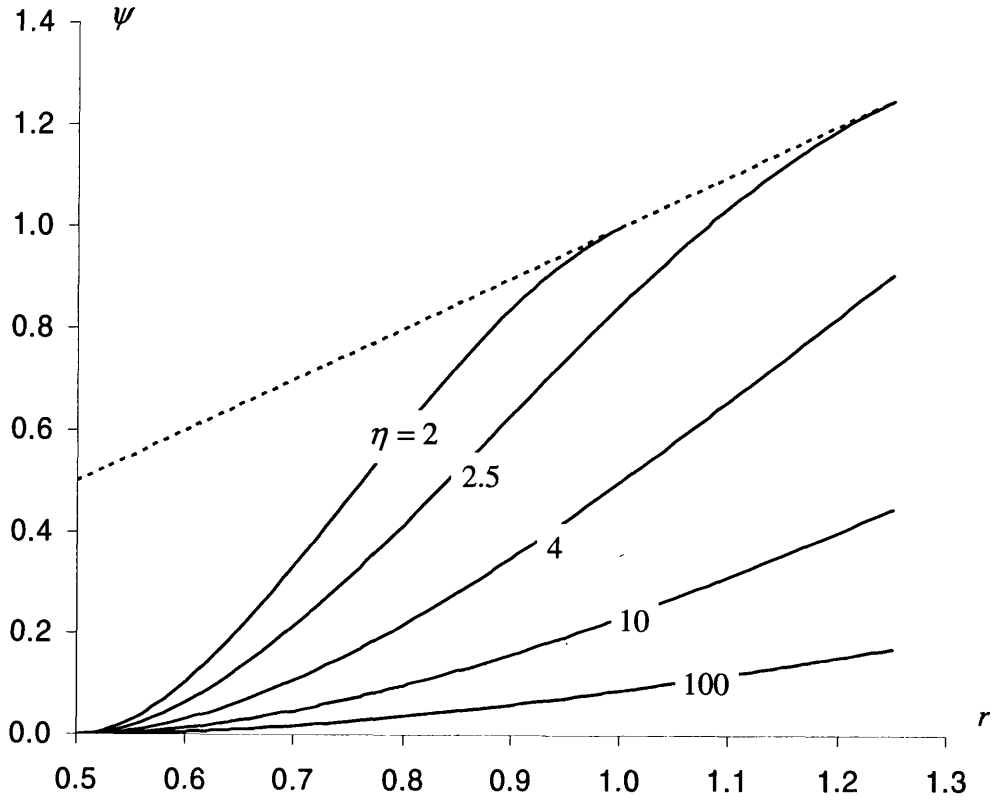


Figure E.1 Section through plate at $\theta = 0$ for increasing values of η. $a = 0.5$

References

[1] WILLIAMS, C., 2000. The definition of curved geometry for widespan enclosures. Proceedings of the *International Symposium on Widespan Enclosures,* 26-28 April 2000, University of Bath, pp.19-26.

[2] FARIN, G. 2002. A history of curves and surfaces in CAGD, in *Handbook of computer-aided geometric design*, ed. G. Farin, J. Hoschek, and M.-S. Kim: Elsevier. pp. 1-22.

[3] ARSLAN, S., FISHER, A., and WILLIAMS, C., 2005. Biomimesis and the geometric definition of shell structures in architecture. Proceedings of the *8th International Conference on Generative Art,* 15-17 December 2005, Politecnico di Milano University, Milan

[4] SEDERBERG, T.W., 2005. *An introduction to B-Spline curves.* [online]. Available from: tom.cs.byu.edu/~455/bs.pdf [Accessed 7 July 2007].

[5] FARIN, G., 2002. *Curves and surfaces for CAGD: a practical guide.* 5th ed. San Francisco: Morgan Kaufmann Publishers.

[6] NEEDHAM, T., 1997. *Visual complex analysis,* Oxford: Clarendon Press.

[7] NICHOLL, C., 2004. *Leonardo Da Vinci,* Penguin Books.

[8] WHITE, F.M., 1991. *Viscous fluid flow.* 2nd ed. McGraw-Hill.

[9] DA VINCI, L., *c.*1508-10. *Sheet of studies of water flowing past obstacles and a sketch of a whirlpool,* The Royal Library, Windsor Castle.

[10] STAM, J., 2003. Real-time fluid dynamics for games. Proceedings of the *Game Developers' Conference,* 4-8 March 2003, San Jose, California, USA

[11] STAM, J., 2003. Flows on surfaces of arbitrary topology. Proceedings of the *SIGGRAPH Annual Conference on Computer Graphics 2003, ACM Transactions On Graphics (TOG),* pp.724-731.

[12] GOODIER, J.N., 1934. An analogy between the slow motions of a viscous fluid in two dimensions and systems of plane stress. *Philosophical magazine.* 17 pp. 554-574.

[13] RICHARDS, T.H., 1960. Analogy between the slow motion of a viscous fluid and the extension and flexure of plates: a geometric demonstration by means of moiré fringes. *British Journal of Applied Physics.* 11 pp. 244-254.

[14] LANGLOIS, W.E., 1964. *Slow viscous flow,* New York: Macmillian Company.

[15] DEAN, W.R., 1958. An application in hydrodynamics of the Greens's function of an elastic plate. *Mathematika.* 5(10) pp. 85-92.

[16] RAYLEIGH, LORD, 1893. On the flow of viscous liquids, especially in two dimensions. *Philosophical Magazine.* 35(Series 5) pp. 354-372.

[17] CHOI, K.-J. and KO, H.-S., 2002. Stable but responsive cloth. *ACM Transactions on Graphics*. 21(3) pp. 604-611.

[18] BÉZIER, P., 1972. *Numerical control: mathematics and applications[Emploi des mechines à commande numerique],* translated by A.R. Forrest and A.F. Pankhurst. London: John Wiley and Sons.

[19] DE CASTELJAU, P., 1959. Outillages méthodes calcul. Technical report, A. Citroën.

[20] DE CASTELJAU, P., 1963. Courbes et surfaces à pôles. Technical report, A. Citroën.

[21] BERNSTEIN, S., 1912. Démonstration du théoreme de Weierstrass fondeé sur le calcul des probabilités. *Harkov Soobs. Matem ob-va*. 13(1-2).

[22] GORDON, W. and RIESENFELD, R., 1974. B-spline curves and surfaces. *Computer Aided Geometric Design*. pp. 95-126.

[23] DE BOOR, C., 1972. On calculating with B-splines. *Journal of Approximation Theory*. 6(1) pp. 50-62.

[24] DE BOOR, C., 1962. Bicubic spline interpolation. *Journal of Mathematical Physics*. 41(2) pp. 212-218.

[25] FERGUSON, J., 1964. Multivarible curve interpolation. *J ACM*. 11(2) pp. 221-228.

[26] HARDER, R.L. and DESMARAIS, R.N., 1972. Interpolation using surface splines. *Journal of Aircraft*. 9(2) pp. 189-191.

[27] MEINGUET, J., 1979. Multivariate interpolation at arbitrary points made simple. *Journal of Applied Mathematics and Physics*. 30 pp. 292-304.

[28] APPA, K., 1989. Finite-surface splines. *Journal of Aircraft*. 26(5) pp. 495-496.

[29] SARRAGA, R.F., 2000. A variational method to model G^1 surfaces over triangular meshes of arbitrary topology in R^3. *ACM Transactions on Graphics*. 19(4) pp. 279-301.

[30] WANG, J.X. and WANG, R.H., 2004. Spline interpolation from random data. *Journal of Information & Computational Science*. 1(1) pp. 121-125.

[31] KNUDSON, W. and NAGY, D., 1974. Spline interpolation and automatic generation of initial geometry for cable-net structures. Proceedings of the *International Conference on Tension Roof Structures,* April 8-10 1974, London, England

[32] DEROSE, T., KASS, M., and TRUONG, T., 1998. Subdivision surfaces in character animation. Proceedings of the *SIGGRAPH Annual Conference on Computer Graphics,* 19-24 July 1998, Orlando, Florida, USA, pp.85-94.

[33] STAM, J., 2001. On subdivision schemes generalizing uniform B-spline surfaces of arbitrary degree. *Computer Aided Geometric Design*. 18(5) pp. 383-396.

[34] DE RHAM, G., 1956. Sur une courbe plane. *Journal de Mathématiques Pures et Appliquées*. 39 pp. 25-42.

[35] CHAIKIN, G., 1974. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*. 3 pp. 346-349.

[36] RIESENFELD, R.F., 1975. On Chaikin's algorithm. *Computer Graphics and Image Processing*. 4 pp. 304-310.

[37] PETERS, J. and REIF, U., 1997. The simplest subdivision scheme for smoothing polyhedra. *Association for Computing Machinery Transactions on Graphics*. 16(4) pp. 420-431.

[38] LOOP, C.T., 1987. *Smooth subdivision surfaces based on triangles.* Thesis(MSc). Department of Mathematics, University of Utah.

[39] ZORIN, D. and SCHRÖDER, P., 2000. *Subdivision for modeling and animation*. [online]. Available from: http://www.multires.caltech.edu/pubs/sig00notes.pdf [Accessed 21 July 2005].

[40] BOEHM, W., 1984. Calculating with box splines. *Computer Aided Geometric Design*. 1(2) pp. 149-162.

[41] BOEHM, W., 1985. Triangular spline algorithms. *Computer Aided Geometric Design*. 2(1-3) pp. 61-67.

[42] BOEHM, W., 1986. Multivariate spline methods in CAGD. *Computer-Aided Design*. 18(2) pp. 102-104.

[43] CATMULL, E. and CLARK, J., 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*. 10(6) pp. 350-355.

[44] DOO, D. and SABIN, M., 1978. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*. 10(6) pp. 356-360.

[45] DYN, N., LEVIN, D., and GREGORY, J.A., 1990. Butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*. 9(2) pp. 160-169.

[46] ZORIN, D. and SCHRÖDER, P., 1996. Interpolating subdivision schemes for meshes with arbitrary topology. Proceedings of the *SIGGRAPH Annual Conference on Computer Graphics and Interactive Techniques*, pp.189-192.

[47] KOBBELT, L., 2000. $\sqrt{3}$-subdivision. Proceedings of the *SIGGRAPH Annual Conference on Computer Graphics and Interactive Techniques,* 23-28 July 2000, New Orleans, Louisiana, USA, pp.103-112.

[48] LABISK, U. and GREINER, G., 2000. Interpolatory $\sqrt{3}$ subdivision. *Computer Graphics Forum (Eurographics 2000)*. 19(3) pp. 131-138.

[49] LI, G., MA, W., and BAO, H., 2004. $\sqrt{2}$ subdivision for quadrilateral meshes. *The Visual Computer*. 20(2-3) pp. 180-198.

[50] HABIB, A. and WARREN, J., 1999. Edge and vertex insertion for a class of C1 subdivision surfaces. *Computer Aided Geometric Design*. 16 pp. 223-247.

[51] STAM, J. and LOOP, C., 2003. Quad/triangle subdivision. *Computer Graphics Forum*. 22(1) pp. 1-7.

[52] STADLER, M. and HOLZAPFEL, G.A., 2004. Subdivision schemes for smooth contact surfaces of arbitrary mesh topology in 3D. *International Journal for Numerical Methods in Engineering*. 60 pp. 1161-1195.

[53] MA, W., 2005. Subdivision surfaces for CAD - an overview. *Computer-Aided Design*. 37(7) pp. 693-709.

[54] SCHRÖDER, P., 2002. Subdivision as a fundamental building block of digital geometry processing algorithms. *Journal of Computational and Applied Mathematics*. 149 pp. 207-219.

[55] COONS, S., 1964. Surfaces for computer aided design. Technical report, MIT.

[56] COONS, S., 1968. Rational bicubic surface patches. Technical report, MIT.

[57] FORREST, A.R., 1968. *Curves and surfaces for computer-aided design.* Thesis(PhD). Cambridge.

[58] JENA, M.K., SHUNMUGARAJ, P., and DAS, P.C., 2003. A non-stationary subdivision scheme for generalizing trigonometric spline surfaces to arbitrary meshes. *Computer Aided Geometric Design*. 20(2) pp. 61-77.

[59] MORIN, G., WARREN, J., and WEIMER, H., 2001. A subdivision scheme for surfaces of revolution. *Computer Aided Geometric Design*. 18(5) pp. 483-502.

[60] YING, L. and ZORIN, D., 2001. Nonmanifold subdivision. Proceedings of the *Conference on Visualization '01,* San Diego, California, pp.325-332.

[61] CHANG, Y.-S., MCDONNELL, K.T., and QIN, H., 2002. A new solid subdivision scheme based on box splines. Proceedings of the *Seventh ACM Symposium on Solid Modeling and Applications SM'02,* June 17-21 2002, Saarbrucken, Germany, pp.226-233.

[62] CHANG, Y.-S. and QIN, H., 2006. A unified subdivision approach for multi-dimensional non-manifold modeling. *Computer-Aided Design.* 38(7) pp. 770-785.

[63] WEIMER, H. and WARREN, J., 1999. Subdivision schemes for fluid flow. Proceedings of the *SIGGRAPH Annual Conference on Computer Graphics,* 8-13 August 1999, Los Angeles, California, USA, pp.111-120.

[64] NG, H.N. and GRIMSDALE, R.L., 1996. Computer graphics techniques for modeling cloth. *IEEE Compter Graphics and Applications.* pp. 28-41.

[65] WEIL, J., 1986. The synthesis of cloth objects. *Computer Graphics and Interactive Techniques (ACM).* 20(4) pp. 49-54.

[66] TERZOPOULOS, D., PLATT, J., BARR, A., and FLEISCHER, K., 1987. Elastically deformable models. Proceedings of the *SIGGRAPH Annual Conference on Computer Animation,* 27-31 July 1987, Anaheim, California, USA, pp.205-214.

[67] ROBERTSON, B., *Meet Geri: the new face of animation*, in *Computer Graphics World.* 1998.

[68] BRIDSON, R., FEDKIW, R., and ANDERSON, J., 2002. Robust treatment of collisions, contact and friction for cloth animation. *Association for Computing Machinery: Proceedings of ACM SIGGRAPH 2002.* 21(3) pp. 594-603.

[69] BARAFF, D., WITKIN, A., and KASS, M., 2003. Untangling cloth. *ACM Transactions on Graphics.* 22(3) pp. 862-870.

[70] CHOI, K.-J. and KO, H.-S., 2005. Research problems in clothing simulation. *Computer-Aided Design.* 37(6) pp. 585-592.

[71] IRVING, G., TERAN, J., and FEDKIW, R., 2004. Invertible finite elements for robust simulation of large deformation. Proceedings of the *Eurographics Symposium on Computer Animation,* 27-29 August 2004, Grenoble, France, pp.131-140.

[72] PALMERIO, B., 1994. An attractive-repulsion mesh adaption model for flow solution on unstructured grids. *Computation and Fluids.* 23(3) pp. 487-506.

[73] BRIDSON, R., MARINO, S., and FEDKIW, R., 2003. Simulation of clothing with folds and wrinkles. Proceedings of the *2003 ACM SIGGRAPH/Eurographics symposium on Computer animation,* San Diego, California, USA, pp.28-36.

[74] GRINSPUN, E., HIRANI, A.N., DESBRUN, M., and SCHRÖDER, P., 2003. Discrete shells. Proceedings of the *2003 Eurographics Symposium on Computer Animation*, pp.62-67.

[75] GRINSPUN, E., 2005. *A discrete model of thin shells.* [online]. Available from: http://ddg.cs.columbia.edu/SIGGRAPH05/NOTES-small.pdf [Accessed 12 July 2006].

[76] BURGOON, R., GRINSPUN, E., and WOOD, Z., 2006. Discrete shells origami. Proceedings of the *21st International Conference on Computers And Their Applications,* 23-25 March 2006, Seattle, Washington, USA

[77] HUGHES, T.J.R., COTTRELL, J.A., and BAZILEVS, Y., 2005. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering.* 194(39-41) pp. 4135-4195.

[78] FORTIN, M. and TANGUY, P., 1984. A non-standard mesh refinement procedure through node labelling. *International Journal for Numerical Methods in Engineering*. 20 pp. 1361-1365.

[79] KAGAN, P., FISCHER, A., and BAR-YOSEPH, P.Z., 1998. New B-spline finite element approach for geometrical design and mechanical analysis. *International Journal for Numerical Methods in Engineering*. 41(3) pp. 435-458.

[80] KAGAN, P., FISCHER, A., and BAR-YOSEPH, P.Z., 2003. Mechanically based models: Adaptive refinement for B-spline finite element. *International Journal for Numerical Methods in Engineering*. 57(8) pp. 1145-1175.

[81] NIKISHKOV, G.P., 2005. Finite element algorithm with adaptive quadtree-octree mesh refinement. *Australian and New Zealand Industrial and Applied Mathematics Journal*. 46(E) pp. C15-C28.

[82] HAMPSHIRE, J.K., TOPPING, B.H.V., and CHAN, H.C., 1992. Three node triangular bending elements with one degree of freedom per node. *Engineering Computations*. 9(1) pp. 49-62.

[83] ZIENKIEWICZ, O.C. and TAYLOR, R.L., 2000. *The finite element method. Volume 1: the basics*. 5th ed. Oxford: Butterworth-Heinemann.

[84] CIRAK, F., ORTIZ, M., and SCHRÖDER, P., 2000. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering*. 47(12) pp. 2039-2072.

[85] STAM, J., 1999. *Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values*. [online]. Available from: http://www.dgp.toronto.edu/people/stam/reality/Research/pdf/cc.pdf [Accessed 20 March 2006].

[86] STAM, J., 1999. *Evaluation of Loop subdivision surfaces*. [online]. Available from: http://www.dgp.toronto.edu/people/stam/reality/Research/pdf/loop.pdf [Accessed 20 March 2006].

[87] IRONS, B.M.R. and DRAPER, K.J., 1965. Inadequacy of nodal connections in a stiffness solution for plate bending. *American Institute of Aeronautics and Astronautics*. 3(5) pp. 961.

[88] REIF, U. and SCHRÖDER, P., 2001. Curvature integrability of subdivision surfaces. *Advances in Computational Mathematics*. 12(2) pp. 157-174.

[89] CIRAK, F. and ORTIZ, M., 2001. Fully C^1-conforming subdivision elements for finite deformation thin-shell analysis. *International Journal for Numerical Methods in Engineering*. 51(7) pp. 813-833.

[90] GOLDSTEIN, H., 1980. *Classical mechanics*. 2nd ed. Addison-Wesley Publishing Company, Inc.

[91] GREEN, S. and TURKIYYAH, G., 2004. Second-order accurate constraint formulation for subdivision finite element simulation of thin shells. *International Journal for Numerical Methods in Engineering*. 61(3) pp. 380-405.

[92] GREEN, S. and TURKIYYAH, G., 2005. A rotation-free quadrilateral thin shell subdivision finite element. *Communications in Numerical Methods in Engineering*. 21(12) pp. 757-767.

[93] DONDL, P.W., SHEN, C.-P., and BHATTACHARYA, K., 2007. Computational analysis of martensitic thin films using subdivision surfaces. *International Journal for Numerical Methods in Engineering*. 72(1) pp. 72-94.

[94] VOLINO, P. and MAGNENAT-THALMANN, N., 2004. Animating complex hairstyles in real-time. Proceedings of the *ACM symposium on Virtual Reality Software and Technology*, 10-12 November 2004, Hong Kong, pp.41-48.

[95] MONAGHAN, J.J., 1982. An introduction to SPH. *Computer Physics Communications*. 48 pp. 89-96.

[96] KIPFER, P. and WESTERMANN, R., Realistic and interactive simulation of rivers.

[97] BELYTSCHKO, T. and TABBARA, M., 1996. Dynamic fracture using element-free Galerkin methods. *International Journal for Numerical Methods in Engineering.* 39(6) pp. 923-938.

[98] WHITE, F.M., 1999. *Fluid mechanics.* 4th ed. McGraw-Hill.

[99] CHUNG, T.J., 2002. *Computational fluid dynamics,* Cambridge University Press.

[100] MASSEY, B. and WARD-SMITH, J., 1998. *Mechanics of fluids.* 7th ed. Stanley Thornes Ltd.

[101] ZIENKIEWICZ, O.C. and TAYLOR, R.L., 2000. *The finite element method. Volume 3: fluid dynamics.* 5th ed. Oxford: Butterworth-Heinemann.

[102] ALFONSI, G. and GIORGINI, A., 2002. Temporal evolution of high-order vortices in the nonsymmetric wake past a circular cylinder. *Fluid Dynamics Research.* 31(1) pp. 13-39.

[103] LESNIC, D., ELLIOTT, L., and INGHAM, D.B., 1998. The boundary element solution of the Laplace and biharmonic equations subjected to noisy boundary data. *International Journal for Numerical Methods in Engineering.* 43(3) pp. 479-492.

[104] DRAKE, R. and MANORANJAN, V.S., 1998. A method of dynamic mesh adaption. *International Journal for Numerical Methods in Engineering.* 39(6) pp. 939-949.

[105] WANSOPHARK, N. and DECHAUMPHAI, P., 2002. Enhancement of streamline upwinding finite element of solutions by adaptive meshing technique. *Japan Society of Mechanical Engineers International Journal, Series B: Fluids and Thermal Engineering.* 45(4) pp. 770-779.

[106] KRYSL, P., GRINSPUN, E., and SCHRÖDER, P., 2003. Natural hierarchical refinement for finite element methods. *International Journal for Numerical Methods in Engineering.* 56(8) pp. 1109-1124.

[107] MÜLLER, K. and HAVEMANN, S., 2000. Subdivision surface tesselation on the fly using a versatile mesh data structures. Proceedings of the *21st Annual Conference of Eurographics,* 21-25 August 2000, Interlaken, Switzerland, pp.C-151-C-159.

[108] AMRESH, A., FARIN, G., and RAZDAN, A. 2003. Adaptive subdivision schemes for triangular meshes, in *Hierarchical and geometric methods in scientific visualization*, ed. G. Farin, H. Hagen, and B. Hamann: Springer-Verlag. pp. 319-327.

[109] SCHROEDER, W.J., ZARGE, J.A., and LORENSEN, W.E., 1992. Decimation of triangle meshes. Proceedings of the *SIGGRAPH Annual Conference on Computer Graphics and Interactive Techniques,* 26-31 July 1992, Chicago, Illinois, USA, pp.65-70.

[110] VIEIRA, A., VELHO, L., LOPES, H., TAVARES, G., and LEWINER, T., 2003. Fast stellar mesh simplification. Proceedings of the *Brazilian Symposium on Computer Graphics and Image Processing,* 12-15 October 2003, São Carlos, Brazil, pp.27-34.

[111] CIRAK, F., SCOTT, M.J., ANTONSSON, E.K., ORTIZ, M., and SCHRÖDER, P., 2002. Integrated modeling, finite element analysis and engineering design for thin-shell structures using subdivision. *Computer-Aided Design.* 34(2) pp. 137-148.

[112] YU, T.Y. and SONI, B.K., 1995. Application of NURBS in numerical grid generation. *Computer-Aided Design.* 27(2) pp. 147-157.

[113] MELESHKO, V.V., 2003. Selected topics in the history of the two-dimensional biharmonic problem. *Applied Mechanics Reviews.* 56(1) pp. 33-85.

[114] THOMPSON, D., 1961. *On growth and form,* Cambridge: University Press.

[115] GREEN, A.E. and ZERNA, W., 1968. *Theoretical elasticity.* 2nd ed. Oxford: Clarendon Press.

[116] FUNG, Y.C., 1965. *Foundations of solid mechanics,* London: Prentice-Hall, Inc.

[117] CALLADINE, C.R., 1983. *Theory of shell structures,* Cambridge: University Press.

[118] GHEORGHIU, A. and DRAGOMIR, V., 1978. *Geometry of structural forms,* London: Applied Science Publishers Ltd.

[119] WOODWARD, L.M. 1986. Differential forms in the theory of surfaces, in *The mathematics of surfaces*, ed. J.A. Gregory, Oxford: Clarendon Press. pp. 95-115.

[120] STRUIK, D.J., 1988. *Lectures on classical differential geometry.* 2nd ed. New York: Dover Publications, Inc.

[121] NUTBOURNE, A.W. 1986. A circle diagram for local differential geometry, in *The mathematics of surfaces*, ed. J.A. Gregory, Oxford: Clarendon Press. pp. 57-71.

[122] MISNER, C.W., THORNE, K.S., and WHEELER, J.A., 1973. *Gravitation,* San Francisco: W. H. Freeman and Company.

[123] ZIENKIEWICZ, O.C. and TAYLOR, R.L., 2000. *The finite element method. Volume 2: solid mechanics.* 5th ed. Oxford: Butterworth-Heinemann.

[124] MASE, G.E., 1970. *Theory and problems of continuum mechanics,* Schaum's outline series in engineering McGraw-Hill.

[125] ODEN, J.T., 1972. *Finite elements of nonlinear continua,* McGraw-Hill Book Company.

[126] TIMOSHENKO, S. and GOODIER, J.N., 1951. *Theory of elasticity.* 2nd ed. McGraw-Hill Book Company.

[127] VERSTEEG, H.K. and MALALASEKERA, W., 2007. *An introduction to computational fluid dynamics. The finite volume method.* 2nd ed. Pearson Prentice Hall.

[128] SIMMONDS, J.G., 1982. *A brief on tensor analysis.* 2nd ed. New York: Springer-Verlag.

[129] WARREN, J. and WEIMER, H., 2002. *Subdivision methods for geometric design: a constructive approach,* San Francisco: Morgan Kaufmann Publisher.

[130] SCHEY, H.M., 1997. *Div, grad, curl and all that: an informal text on vector calculus.* 3rd ed. New York: W. W. Norton & Company.

[131] ZIENKIEWICZ, O.C., 1977. *The finite element method.* 3rd ed. London: McGraw-Hill Book Company Ltd.

[132] WILLIAMS, C. 1986. Defining and designing curved flexible tensile surface structures, in *The mathematics of surfaces*, ed. J.A. Gregory, Oxford: Clarendon Press. pp. 143-177.

[133] ROCKEY, K.C., EVANS, H.R., GRIFFITHS, D.W., and NETHERCOT, D.A., 1975. *The finite element method,* London: William Clowes and Sons, Ltd.

[134] SHEWCHUK, J., *Triangle: A two-dimensional quality mesh generator and Delaunay triangulator.* [online]. Available from: http://www.cs.cmu.edu/~quake/triangle.html [Accessed 30 July 2007].

[135] SHEWCHUK, J.R. 1996. Triangle: engineering a 2D quality mesh generator and Delaunay triangulator, in *Applied computational geometry: towards geometric engineering, from the First ACM Workshop on Applied Computational Geometry*, ed. M.C. Lin and D. Manocha, Berlin: Springer-Verlag. pp. 203-222.

[136] SHEWCHUK, J.R., 2002. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications.* 22(1-3) pp. 21-74.

[137] DIRAC, P.A.M., 1996. *General theory of relativity,* Chichester: Princeton University Press.

[138] OTTO, F., 1995. *Frei Otto, Bodo Rash: finding form. Towards an architecture of the minimal.,* Deutscher Werkbund Bayern.

[139] WHALLEY, A., 2000. The eden project glass houses world environments. Proceedings of the *International Symposium on Widespan Enclosures,* 26-28 April 2000, University of Bath, pp.39-48.

[140] DAY, A.S., 1965. An introduction to dynamic relaxation. *The Engineer.* 219 pp. 218-221.

[141] BARNES, M.R., 1999. Form finding and analysis of tension structures by dynamic relaxation. *International Journal of Space Structures.* 14(2) pp. 89-104.

[142] SOUTHWELL, R.V., 1958. Use of an analogue to resolve Stokes's paradox. *Nature, Letter to the Editor.* 181(4618) pp. 1257-1258.

[143] JAEGER, L.G., 1964. *Elementary theory of elastic plates,* Pergamon Press.

[144] LOWE, P.G., 2005. *Basic principles of plates and slabs,* Caithness: Whittles Publishing Ltd.

[145] TIMOSHENKO, S. and WOINOWSKY-KRIEGER, S., 1959. *Theory of plates and shells.* 2nd ed. McGraw-Hill Book Company.

[146] HU, Z.Z., 2000. *Numerical simulation of laminar separated flows on adaptive tri-tree grids with the finite volume method.* Thesis(PhD). Department of Mechanical Engineering, University College London.

[147] PRANDTL, L. and TIETJENS, O.G., 1957. *Applied hydro- & aeromechanics,* New York: Dover.

[148] TANEDA, S., 1956. Experimental investigation of the wakes behind cylinders and plates at low Reynolds numbers. *Journal of the Physical Society of Japan*. 11(3) pp. 302-307.

[149] DING, Y. and KAWAHARA, M., 1999. Three-dimensional linear stability analysis of incompressible viscous flows using the finite element method. *International Journal for Numerical Methods in Fluids*. 31(2) pp. 451-479.

[150] KÁRMÁN, T., VON, 1911. Über den Mechanismus des Widerstandes, den ein bewegter Körper in einer Flüssigkeit erzeugt. *Nachr. Ges. Wiss. Göttingen Math.-Phys.* pp. 509-517, 547-556.

[151] FRANKE, R., RODI, W., and SCHÖNUNG, B., 1990. Numerical calculation of laminar vortex-shedding flow past cylinders. *Journal of Wind Engineering and Industrial Aerodynamics*. 35(1) pp. 237-257.

[152] BRAZA, M., CHASSAING, P., and HA MINH, H., 1986. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journal of Fluid Mechanics*. 165 pp. 79-130.

[153] KARANTH, D., RANKIN, G.W., and SRIDHAR, K., 1995. Computational study of flow past a cylinder with combined in-line and transverse oscillation. *Computational Mechanics*. 16(1) pp. 1-10.

[154] ROSHKO, A., 1955. On the wake and drag of bluff bodies. *Journal of Aeronautical Sciences*. 22(2) pp. 124-132.

[155] BRAZA, M., CHASSAING, P., and HA MINH, H., 1990. Prediction of large-scale transition features in the wake of a circular cylinder. *Physics of Fluids A: Fluid Dynamics*. 2(8) pp. 1461-1471.

[156] COUTANCEAU, M., BOYCE, P., and GUERINEAU, G., 1988. Sur le rôle des décollements d'ordre supérieur sur la naissance de l'instabilité secondaire de la frontière du sillage d'un cylindre circulaire (*On the role of high order separations on the onset of the secondary instability of the circular cylinder wake boundary*). *Comptes rendus de l'Académie des sciences. Série 2*. 306(18) pp. 1259-1263.

[157] COUTANCEAU, M. and DEFAYE, J.-R., 1991. Circular cylinder wake configurations: a flow visualization survey. *Applied Mechanics Review.* 44(6) pp. 255-305.

[158] TOPPING, B.H.V., SZIVERI, J., BAHREINEJAD, A., LEITE, J.P.B., and CHENG, B., 1998. Parallel processing, neural networks and genetic algorithms. *Advances in Engineering Software.* 29(10) pp. 763-786.

[159] TOPPING, B.H.V. and KHAN, A.I., 1994. Parallel computation methods for dynamic relaxation. *Engineering Computations.* 11 pp. 513-548.

[160] PIPONI, D. and BORSHUKOV, G.D., 2000. Seamless texture mapping of subdivision surfaces by model pelting and texture blending. Proceedings of the *SIGGRAPH Annual Conference on Computer Graphics and Interactive Techniques,* 23-28 July 2000, New Orleans, Louisiana, USA, pp.471-478.

[161] ELSHABKA, A.M. and CHUNG, T.J., 1999. Numerical solution of three-dimensional stream function vector components of vorticity transport equations. *Computer Methods in Applied Mechanics and Engineering.* 170(1-2) pp. 131-153.

[162] GREYWALL, M.S., 1993. Streamwise computation of three-dimensional flows using two stream functions. *Journal of Fluids Engineering, Transactions of the ASME.* 115(2) pp. 233-238.

[163] SCHAEFER, S., HAKENBERG, J., and WARREN, J., 2004. Smooth subdivision tetrahedral meshes. Proceedings of the *Eurographics Symposium on Graphics Processing,* July 2004, Nice, France, pp.151-158.